

GE Fanuc Automation

Speicherprogrammierbare Steuerungen

PC Control Software

GFK-1424B-GE

August 1998

Die Begriffe Vorsicht, Achtung und Hinweis, wie sie in dieser Publikation verwendet werden

Vorsicht

In dieser Veröffentlichung werden VORSICHT-Hinweise verwendet, um darauf hinzuweisen, daß innerhalb der beschriebenen Geräte gefährliche Spannungen, Ströme, Temperaturen oder andere Bedingungen, die körperliche Schäden hervorrufen können, vorkommen.

Wo Unaufmerksamkeit entweder körperliche Schäden oder eine Beschädigung des Geräts verursachen können, werden VORSICHT-Hinweise verwendet.

Achtung

ACHTUNG-Hinweise werden dort verwendet, wo das Gerät bei unsachgemäßer Vorgehensweise beschädigt werden könnte.

Hinweis

HINWEISE sollen nur die Aufmerksamkeit des Lesers auf Informationen lenken, die besonders wichtig für Verständnis und Bedienung des Geräts sind.

Dieses Dokument stützt sich auf Informationen, die zum Zeitpunkt seiner Veröffentlichung verfügbar waren. Obwohl alle Anstrengungen unternommen wurden, den Inhalt so genau wie möglich zu gestalten, können die hier enthaltenen Informationen nicht den Anspruch erheben, alle Details oder Veränderungen von Software und Hardware abzudecken, oder jede Möglichkeit im Zusammenhang mit Installation, Betrieb oder Wartung zu berücksichtigen. In diesem Dokument können Merkmale beschrieben sein, die nicht in allen Hard- und Softwaresystemen vorhanden sind. GE Fanuc Automation übernimmt keine Verpflichtung, Besitzer dieses Dokuments über nachträglich durchgeführte Änderungen zu informieren.

GE Fanuc Automation übernimmt keine Verantwortung für Genauigkeit, Vollständigkeit oder Nützlichkeit der in diesem Dokument enthaltenen Informationen.

Bei den folgenden Bezeichnungen handelt es sich um Warenzeichen für Produkte GE Fanuc Automation North America, Inc.

Alarm Master	GEnet	PowerMotion	Series One
CIMPLICITY	Genius	ProLoop	Series Six
CIMPLICITY PowerTRAC	Genius PowerTRAC	PROMACRO	Series Three
CIMPLICITY 90–ADS	Helpmate	Series Five	VuMaster
CIMSTAR	Logicmaster	Series 90	Workmaster
Field Control	Modelmaster		

Änderungen in diesem Handbuch

Ausgabestand 2.0 der PC Control Software enthält folgende neue Funktionen:

Meldung von Laufzeitfehlern — Seite 2-5

Notautorisierung — Seite 2-14

Intelligente USV-Abschaltüberwachung — Seiten 2-19 und 2-30

Neues Dialogfenster “Systemoptionen” — Seite 2-26

Struktur- und Funktionsblocknumerierung — Seite 2-29

IEC-Referenzanzeige — Seite 2-29

Import/Export von Symbolen über eine .CSV-Datei für globale Symbole und GE Fanuc PCIM und PCIM-Variable — Seite 3-41

Erweiterte Syntaxprüfung — Seite 4-102

Nahtlose Onlinebearbeitung — Seite 4-115

Programmierung von autonomen Programmen in strukturiertem Text und Anweisungsliste — Abschnitte 4 und 5 in Kapitel 4

Menügesteuerter Editor für strukturierten Text — Abschnitt 4 in Kapitel 4

ActiveX-Schnittstelle — Seite 6-33

PCIF2-Treiberunterstützung für Series 90-30 E/A-Schnittstelle — Anhang C

APM- und DSM-Konfiguration; Programm 0 erstellen und herunterladen — Seite E-2 und Onlinehilfe zur Konfiguration Ihrer Bewegungssteuerungskarte

Motion Control Programmierung und Bedienerschnittstellen-Bedienelemente — Anhang E

Unterstützung von indirekter Adressierung (Pointer) — Anhang F

Bedienerschnittstellenfunktion wechseln — “Verschiedene Funktionen” in Anhang G

Zusätzliche Timertypen — “Erweiterte Timer” in Anhang G

Neue Anweisungen für Konvertierung und erweiterte Timer — Anhang G

Unterstützung anwenderdefinierter C-Funktionsblöcke — Einzelheiten zur Benutzung des CFB-Editors siehe Anwendungshinweis APP981R1 auf PC Control CD

Vorwort

Inhalt dieses Handbuches

Kapitel 1. Einleitung: Voraussetzungen und Vorgehensweise bei der Installation von PC Control auf Ihrem Computer, sowie Benutzungshinweise zur Online-Dokumentation.

Kapitel 2. Erste Schritte: Überblick über die Arbeitsweise von PC Control und die wichtigsten Schritte zum Erstellen eines Steuerungsprogramms. Erstellen eines neuen Projektes zur Aufnahme aller zusammenhängender Dateien.

Kapitel 3. I/O-Konfiguration: Beschreibt die Konfiguration der GENIUS® E/A und der Series 90™-30 E/A im Rahmen der PC Control Software.

Kapitel 4. Erstellen von Anwenderprogrammen: Erstellen von Ablaufplänen, Kontaktplanlogik, strukturiertem Text und Anweisungslisten-Programmen.

Kapitel 5. Ablauf von Anwenderprogrammen: Beschreibt das Starten der Runtime-Systeme und wie man ein Anwenderprogramm laufen läßt.

Kapitel 6. Bedienerschnittstellen-Anwendungen erstellen: Erstellung eines Bedienerschnittstellenmenüs mit dem integrierten PC Control BOF-Editor.

Anhang A. Das Personalcomputer-Schnittstellenmodul (PCIM) für Genius E/A: Installation des PCIM in Ihrem Personalcomputer und Konfiguration des Moduls mit dem PCIM Konfigurations-Dienstprogramm in der PC Control Software.

Anhang B. Die Personalcomputer-Schnittstelle (PCIF) für Series 90-30 E/A: Dieser Anhang beschreibt den Einbau der PCIF in Ihren Personalcomputer.

Anhang C. Personalcomputer-Schnittstelle (PCIF2) für Series 90-30 I/O: Dieser Anhang beschreibt den Einbau der PCIF2 in Ihren Personalcomputer.

Anhang D. Anwendungsbeispiel: Schritttweise Anweisungen zur Erstellung eines Anwendungsprogramms mit AS- und KOP-Programmeditoren. Erstellung einer Bedienerschnittstelle zur Überwachung und Steuerung des Programmbetriebs.

Anhang E. Motion Control: Verwendung der RS-274D entsprechenden Sprache textueller Anweisungen zur Bewegungssteuerung.

Anhang F. Pointer: (Für fortgeschrittene Anwender) Einsatz von Pointern zur indirekten Adressierung in strukturiertem Text.

Anhang G. Befehlssätze

Anhang H. Glossar der Fachausdrücke

Zugehörige Veröffentlichungen

GEK-90486-1	*Genius® E/A-System, Anwenderhandbuch*
GFK-0898	*Series 90™-30 Speicherprogrammierbare Steuerungen, E/A-Modul, Technische Daten*
GFK-0826	*Field Control Dezentrales E/A- und STeuerugnssystem, Anwenderhandbuch*
GFK-0356	*Series 90™-30 Speicherprogrammierbare Steuerungen, Installationshandbuch*
GFK-1180	*CIMPLICITY® HMI für Window NT® und Windows® 95 Basissystem, Anwenderhandbuch*

Inhalt

Inhalt

Inhalt

Kapitel 1 Einleitung

Willkommen bei PC Control, der neuen Software von GE Fanuc, einem Paket mit integrierter Programmierung, Konfiguration, graphischer Bedienoberfläche und einem unter Windows NT® 4.0 ablauffähigen Runtimepaket. PC Control unterstützt Genius®, Field Control™, DeviceNet™, PROFIBUS und Series 90™-30 E/A.

Mit PC Control können Sie:

- Ein Anwendungsprojekt erstellen
- Ihre E/A-Hardware konfigurieren
- Symbole erzeugen und editieren
- Ein Anwenderprogramm erstellen und editieren
- Das Programm unter Windows NT laufen lassen

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Anforderungen und Prozeduren für die Installation von PC Control auf Ihrem Computer
- Anleitungen zum Ablauf von PC Control unter Windows® 95
- Wie Sie die Online-Hilfe des Produkts benutzen können

Windows NT und Windows® 95 sind eingetragene Warenzeichen der Microsoft Corporation. DeviceNet ist ein Warenzeichen der Open DeviceNet Vendor Association, Inc.

1

Abschnitt 1: PC Control installieren und einrichten

Dieser Abschnitt beschreibt die für den Ablauf von PC Control auf Ihrem Computer empfohlenen Systemanforderungen und erläutert, wie das Programm installiert wird.

Ehe Sie anfangen

Ablauf unter Windows NT

Damit Sie PC Control für Windows NT erfolgreich installieren und ausführen können, muß Ihr System folgende Mindestanforderungen erfüllen.

Windows NT 4.0 benötigt Service Pack 3.

	Empfohlen
CPU	Pentium, 100 MHz
RAM	32 MB (64 MB bevorzugt)
Festplatte	100 MB freier Speicherplatz
CD-ROM-Laufwerk	Ja
Monitor	VGA Minimum SVGA empfohlen
Tastatur und Zeigergerät	Ja
USV-System	Empfohlen

Es werden geeignete E/A-Schnittstellen benötigt. Zur Unterstützung gleichzeitig ablaufender Anwendungen können zusätzliche Betriebsmittel benötigt werden.

Schauen Sie auch in dem mitgelieferten Dokument *Wichtige Produktinformation* nach, das neueste Änderungen zu diesen Anforderungen enthalten kann.

Ablauf unter Windows 95

Vorsicht

Windows 95 ist kein Betriebssystem mit geschütztem Modus. Wir empfehlen nicht die Steuerung von Prozessen mit PC Control unter Windows 95. Das Windows 95 Betriebssystem sollte nur zur Programmentwicklung verwendet werden.

PC Control unterstützt die Offline-Entwicklung und Demonstrationsbetrieb unter Windows 95, bei dem die E/A in Simulationsmodus ist. Damit PC Control unter Windows 95 ablaufen kann, muß Ihr System folgende Mindestanforderungen erfüllen.

	Empfohlen
CPU	Pentium, 100 MHz
RAM	32 MB (64 MB bevorzugt)
Festplatte	100 MB freier Speicherplatz
CD-ROM-Laufwerk	Ja
Monitor	VGA Minimum SVGA empfohlen
Tastatur und Zeigergerät	Ja
USV-System	Empfohlen

Schauen Sie auch in dem mitgelieferten Dokument "Wichtige Produktinformation" nach, ob sich diese Anforderungen in der Zwischenzeit geändert haben.

PC Control hochrüsten

Wenn Sie eine autorisierte Version von PC Control hochrüsten wollen, führen Sie vor der Installation von Version 2.0 die folgenden Schritte aus.

1. Doppelklicken Sie im Systemsteuerungsmenü auf "Programme hinzufügen/entfernen".
2. Klicken Sie auf die Karteikarte "Programme installieren/deinstallieren".
3. Selektieren Sie PC Control und klicken auf "Hinzufügen/entfernen".
4. Klicken Sie auf "Ja", um frühere Versionen zu entfernen.
5. Löschen Sie mit dem Explorer den Ordner BIN im Pfad \PCCONTROL\BIN
6. Folgen Sie den Anweisungen zum Installieren und Autorisieren der PC Control Software.

PC Control Software installieren

Ehe Sie anfangen, schauen Sie in dem mitgelieferten Dokument Wichtige Produktinformation, ob es neuere Änderungen oder spezielle Hinweise zur Installation gibt.

Bei der vorliegenden Installationsanweisung wird angenommen, daß das CD-ROM-Laufwerk die Bezeichnung D: hat. Trifft dies für Ihr System nicht zu, müssen Sie die her gegebenen Anweisungen entsprechend abändern.

Sie können die Installation jederzeit abbrechen oder verlassen. Um PC Control für Windows NT zu installieren:

1. Legen Sie die CD-ROM in das CD-ROM-Laufwerk ein. Das Installationsprogramm wird automatisch aktiviert. Doppelklicken Sie auf das Symbol neben Setup.exe. Hierauf erscheint das PC Control Installationsmenü.
2. Lesen Sie die Einführung. Klicken Sie auf Weiter, um mit dem Installationsvorgang fortzufahren.
3. Hierauf erscheint die Dialogbox "Zielverzeichnis auswählen". Das Standardverzeichnis ist C:\CIMPLICITY\PCCONTROL. Um das Standardverzeichnis zu wechseln, klicken Sie auf die Schaltfläche "Suchen in" und wählen ein anderes Verzeichnis. Klicken Sie auf "Weiter", um das Verzeichnis zu übernehmen.
4. Die Dateien von PC Control werden im Zielverzeichnis installiert. Dieser Vorgang kann einige Minuten in Anspruch nehmen.

PC Control Software ausführen und autorisieren

Nachdem Sie PC Control installiert haben, können Sie die Software entweder im Demonstrationsmodus laufen lassen oder sie zum Betrieb im autorisierten Modus registrieren. Der Demonstrationsmodus ist auf 32 E/A-Punkte und eine Betriebsdauer von 2 Stunden beschränkt.

Betrieb von PC Control im Demonstrationsmodus:

1. Suchen Sie im Menü "Programme" das Symbol "Programmeditor":



Hierauf erscheint die Dialogbox "Validierung".

2. Klicken Sie auf die Schaltfläche **Demo**. Hierauf erscheint die Dialogbox "Lizenzinformationen".

3. Lesen Sie die Vereinbarung und klicken Sie auf die Schaltfläche **Einverstanden**, um die Bedingungen anzuerkennen und mit der Installation fortzufahren. Hierauf erscheint die Dialogbox "Passwort eingeben".

4. Geben Sie das Standard-Passwort 4, 5, 6, 7 ein und klicken dann auf die Schaltfläche **Eingabe**.

   PC Control läuft jetzt im Demonstrationsmodus.

Betrieb von PC Control im autorisierten Modus:

Sie können nur gekaufte Software im autorisierten Modus laufen lassen. Demo-CDs können nicht autorisiert werden.

1. Suchen Sie im Menü "Programme" das Symbol "Programmeditor":



   Hierauf erscheint die Dialogbox "Validierung".

2. Klicken Sie auf die Schaltfläche **Autorisierung**. Hierauf erscheint die Dialogbox "Lizenzinformationen".

3. Lesen Sie die Vereinbarung und klicken Sie auf die Schaltfläche **Einverstanden**, um die Bedingungen anzuerkennen und mit der Installation fortzufahren. Hierauf erscheint die Dialogbox "Autorisierung" mit einem Anlagencode.

4. Zur Autorisierung Ihrer Software, folgen Sie den Anweisungen am Bildschirm.

   Geben Sie den Anlagencode ein und klicken dann auf **OK**. Hierauf erscheint die Dialogbox "Passwort eingeben".

5. Geben Sie das Standard-Passwort 4, 5, 6, 7 ein und klicken dann auf die Schaltfläche **Eingabe**.

   PC Control läuft jetzt im autorisierten Modus.

Abschnitt 2: On-Line-Dokumentation verwenden

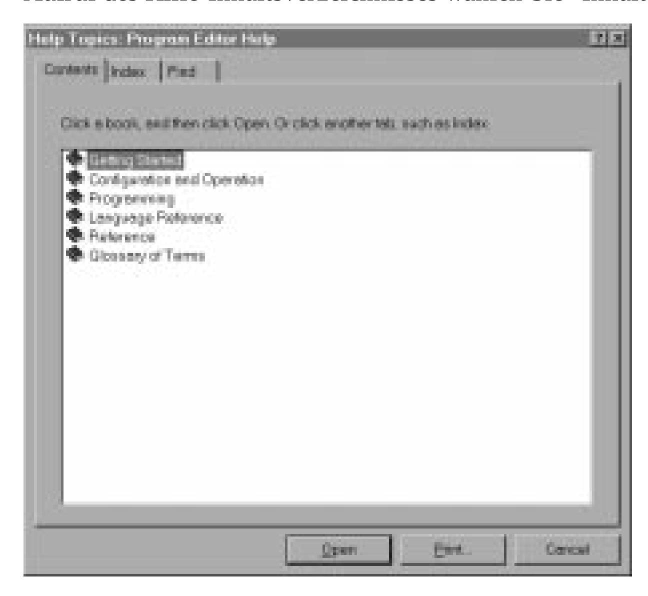
Bei der Arbeit mit PC Control, bildet die Online-Hilfe ihre Haupt-Informationsquelle.. Die Online-Hilfe enthält eine Demo, die Ihnen beim Einstieg in das Produkt helfen soll, sowie spezielle Hilfethemen, die die Produkteigenschaften und deren Benutzung beschreiben.

Online-Hilfe

Die Online-Hilfe ist so aufgebaut, daß Sie schnell und einfach auf die Informationen zu den Themen von PC Control zugreifen können. Verwenden Sie die Hilfe, um allgemeine Informationen zu einer Produkteigenschaft zu erhalten, die Durchführung einer speziellen Prozedur zu erlernen, oder die Definition eines Ausdrucks zu finden, mit dem Sie nicht vertraut sind. Es gibt drei verschiedene Methoden, die Hilfeinformationen zu finden: Das Hilfe-Inhaltsverzeichnis, die Hilfe-Suchfunktion oder die Kontexthilfe.

Hilfe-Inhaltsverzeichnis

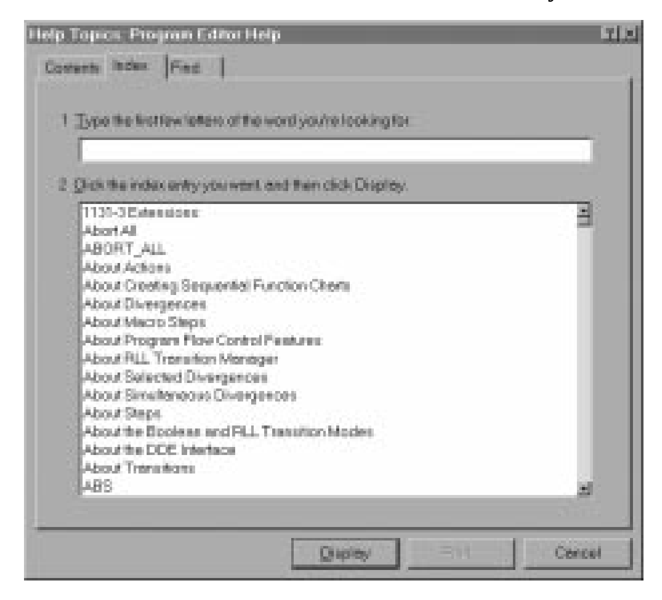
Die Karteikarte "Inhalt" der Hilfefunktion zeigt die Haupt-Hilfethemen. Zum Aufruf des Hilfe-Inhaltsverzeichnisses wählen Sie "Inhalt" im Menü "Hilfe":



Doppelklicken Sie auf die Themenbezeichnung oder das Buchsymbol daneben, um das Buch zu öffnen und die zugehörigen Unterthemen anzuzeigen. Doppelklicken Sie auf einen der Untertitel, um dessen Inhalt anzuzeigen.

Hilfe durchsuchen

Mit der Hilfe-Karteikarte "Index" können Sie das gesuchte Thema entweder eintippen oder eine alphabetisch geordnete Themenliste durchblättern. Zum Aufruf des Hilfeindex wählen Sie im Hilfemenü das Element "Suche Hilfe zu" (oder klicken Sie auf die Schaltfläche "Hilfe-Suchfunktion" in der Symbolleiste):



Um eine Suche anzustoßen, geben Sie entweder die Themenbezeichnung über die Tastatur in das Feld ein oder blättern durch die Liste. Doppelklicken Sie auf das selektierte Thema, um die zugehörigen Hilfeinformationen anzuzeigen.

Kontexthilfe (F1)

Die Kontexthilfe ermöglicht es Ihnen, über die Taste F1 schnell auf bestimmte Informationen zum aktiven Fenster oder der aktiven Dialogbox zuzugreifen.

Kapitel 2 Erste Schritte

Öffnen Sie zu Beginn zunächst das Dienstprogramm "Programmeditor". Im Programmeditor können Sie ein neues Projekt mit Anwenderprogramm und Konfigurationsdateien erstellen, Anwenderprogramme erstellen und das Dienstprogramm "E/A-Konfiguration" aufrufen.

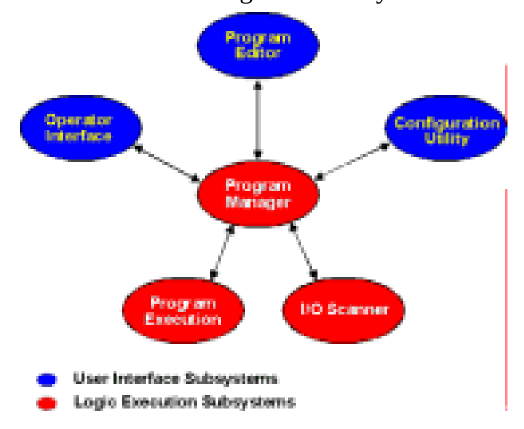
In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Übersicht über die Software-Subsysteme, die die PC Control Software enthalten
- Übersicht über IEC 1131
- Wie Sie Zugriffsebenen-Passworte verändern können
- Wie Sie vom Programmeditor aus Projekte erstellen und verwalten können
- Wie Sie Anwenderprogramm erstellen und verwalten können
- Übersicht über die Arbeitsweise des Programms

2

Abschnitt 1: Übersicht über die Software

Die PC Control Software besteht aus folgenden Subsystemen:



Software-Subsysteme

Konfigurations-Dienstprogramm Mit dem Konfigurations-Dienstprogramm können Sie die E/A-Struktur definieren und E/A-Punkten und E/A-Ports Namensetikette zuweisen.

Bedienerschnittstelle (PC Control GBO) Mit dem Bedienerschnittstellen-Dienstprogramm können Sie die Umgebung der Bedienerschnittstelle kundenspezifisch anpassen und bedienen. Die Bedienerschnittstelle besteht aus einer Reihe von Bildschirmen, Meldungen und Fenstern, die dem Bediener zur Überwachung und Steuerung einer Maschine oder eines Prozesses zur Verfügung gestellt werden.

Programmeditor Mit dem Programmeditor können Sie Programme in Kontaktplan (KOP), Ablaufplan (AS+), strukturiertem Text und Anweisungsliste erstellen.

Programm-Manager Der Programm-Manager bereitet die Programme für den Ablauf vor und verwaltet die externe Schnittstelle zu Programmausführung und E/A-Zyklussteuerung.

Programmausführung Die Programmausführung führt die Programme in Kontaktplanlogik, Ablaufplan (AS+), strukturiertem Text und Anweisungsliste aus. Programmausführung läuft mit Windows NT's REAL_TIME Prozeßpriorität.

E/A-Zyklussteuerung Die E/A-Zyklussteuerung fragt die physikalischen E/A-Geräte ab und gibt die erfaßten Daten weiter an die Programmausführung. Die E/A-Zyklussteuerung läuft mit REAL_TIME Prozeßpriorität. Die E/A-Zyklussteuerung erfaßt alle Eingangsdaten von der physikalischen E/A-Schnittstelle und überträgt alle Ausgangsdaten dorthin.

Übersicht über IEC 1131

IEC 1131-3 ist eine internationale Norm der International Electrotechnical Commission. In IEC 1131-3 werden Syntax und Semantik einer vereinheitlichten Reihe von Programmiersprachen für speicherprogrammierbare Steuerungen festgelegt.

IEC 1131-3 Programmiersprachen

In der IEC-1131-3 gibt es Textsprachen und graphische Sprachen. Diese Sprachen können zusammen in einer integrierten Programmierumgebung eingesetzt werden. Informationen zu den Erweiterungen der PC Control Software gegenüber der IEC 1131-3 Norm finden Sie in der Online-Hilfe.

Textsprachen	
Anweisungsliste (AL)	Das Format der Anweisungsliste ist ähnlich dem Format einer Assemblersprache.
Strukturierter Text (ST)	Strukturierter Text ist am besten geeignet für komplexe Algorithmen, Operationen mit Zeichenfolgen und Dateien, sowie die Manipulation von Datenstrukturen, die vorzugsweise in einer prozeduralen (textorientierten) Sprache durchgeführt werden. Diese Sprache ist bequem für Personen mit Erfahrung in strukturiertem BASIC, Pascal, C oder anderen höheren Programmiersprachen.
Graphiksprachen	
Kontaktplanprogramm (Kontaktplanlogik — KOP)	Kontaktplanprogramme werden in speicherprogrammierbaren Steuerungen häufig zum Aufbau diskreter Programme verwendet. Kontaktplanprogramme sind so aufgebaut, daß sie den Schaltbildern elektrischer Relaislogik ähneln. Das Kontaktplanprogramm enthält zwei vertikale Stromschienen. Die linke Stromschiene wird als die elektrische Stromquelle betrachtet und wird immer dann stromführend, wenn das Programm ausgeführt wird. Die Stromschiene auf der rechten Seite wird als die elektrische Stromsenke betrachtet. Die beiden Stromschienen sind durch horizontale Strompfade miteinander verbunden, in denen die logischen Anweisungen angeordnet sind.
Ablaufsprache (AS)	AS ist am besten geeignet für Maschinenablaufsteuerungen und für Steuerungsanwendungen mit mehreren Betriebsarten (z.B. Handbetrieb und Automatikbetrieb). Ein Ablaufplan stellt ein Anwenderprogramm als eine Reihe aufeinanderfolgender Schritte dar. Schritte werden über Verknüpfungen miteinander verbunden und die Steuerung wird entsprechend der Programmbearbeitung über Transitionen von einem Schritt zum anderen weitergegeben. Die Steuerlogik kann im Schritt plaziert werden oder in einer mit dem Schritt verknüpften Aktion. Die Steuerlogik wird ausgeführt, wenn der Schritt aktiv wird. Die Steuerung geht zum nächsten Schritt weiter, wenn die Transition ausgelöst wird.

Funktionsbausteinsprache (FBS)	Die Funktionsbausteinsprache ist am besten geeignet für Analogsignal-Verarbeitung sowie für kontinuierlich ablaufende Steuerungsprozesse oder -algorithmen. In der FBS erscheinen Funktionen und Funktionsblöcke als graphische Blöcke mit Funktionsnamen und Ein- und Ausgängen. Die Funktionsbausteindiagramme bestehen aus einem Netzwerk dieser über Signale miteinander verbundenen graphischen Funktionen und Funktionsblöcke. Diese Sprache wird von PC Control derzeit nicht unterstützt.

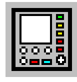
Schnellstart

Hinweis

Um die PC Control Software installieren oder hochrüsten zu können, müssen Sie als Administrator eingetragen sein.

1. Installieren Sie die gesamte PC-Hardware und alle E/A-Module.
2. Starten Sie das Windows NT Betriebssystem.
3. Installieren Sie die PC Control Software und die E/A-Treiber.
4. Erstellen Sie mit dem Dienstprogramm "Programmeditor" ein neues Projekt, das das Anwenderprogramm und die Konfigurationsdateien aufnimmt.
5. Verwenden Sie das Konfigurations-Dienstprogramm, um die Systemhardware zu konfigurieren, und definieren Sie E/A-Symbolname für die Anwenderprogramme. Das Konfigurations-Dienstprogramm wird im Menü "Datei" des Programmeditors gestartet.
6. Bauen Sie das Anwenderprogramm mit dem Dienstprogramm "Programmeditor" auf.
7. Bauen Sie die Bedienerschnittstellenmenüs mit dem Bedienerschnittstellen-Menüeditor (im Menü "Werkzeuge") auf.
8. Testen Sie die Anwenderprogramme mit dem Bedienerschnittstellen- und Programmeditor.
9. Falls gewünscht, können Sie Windows NT so konfigurieren, daß die CIMPLICITY® PC Control Software und die Anwenderprogramme beim Einschalten automatisch gestartet werden.

PC Control Software ausführen



Bei einem Steuerungssystem mit Bedienerschnittstellensoftware:

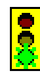
Öffnen Sie im Programm-Manager die Anwendungsgruppe PC Control. Doppelklicken Sie auf das Symbol "Bedienerschnittstelle".

Oder wählen Sie bei laufendem Programmeditor *Werkzeuge* im Programmeditormenü *Bedienerschnittstelle*.

Hinweis

Sind die PC Control Runtime-Subsysteme nicht aktiv, dann können Sie sie starten.

Starten der PC Control Runtime-Subsysteme



1. Öffnen Sie vom Programm-Manager aus die Anwendungsgruppe PC Control.
2. Doppelklicken Sie auf das Symbol PC Control Runtime. Bei aktiven Runtime-Subsystemen erscheint das Runtimesymbol verkleinert unten am Bildschirm.

Bei aktiven Runtime-Subsystemen erscheint das Runtimesymbol unten am Bildschirm auf der Taskleiste oder dem Tablett.

Wird das Runtimesymbol auf dem Tablett angezeigt, müssen Sie die Runtime-Subsysteme mir der Maus schließen. Erscheint die Runtimeanwendung auch in der Taskleiste, kann sie über die Tastatur geschlossen werden. Um das Runtimesymbol in der Taskleiste anzuzeigen, rufen Sie im Menü *Werkzeuge* des Programmeditors das Element *Systemoptionen* auf und überprüfen die Option in der Karteikarte *Anzeigeeigenschaften*.

PC Control Runtime-Subsysteme schließen

1. Klicken Sie die rechte Maustaste auf das Runtimesymbol in der Taskleiste.
2. Wählen Sie Runtime schließen.

Programmeditor starten



1. Öffnen Sie vom Programm-Manager aus die Anwendungsgruppe PC Control.
2. Klicken Sie auf das Programmeditor-Symbol.

PC Control Runtime-Subsysteme

Die Subsysteme Programm-Manager, Programmausführung und E/A-Zyklussteuerung werden optisch dargestellt durch das PC Control Runtimesymbol. Das Ereignisprotokoll hat sein eigenes Symbol. Die Subsysteme bestehen aus folgenden Elementen:

Der Programm-Manager

Der Programm-Manager bereitet die Programme für den Ablauf vor und verwaltet die externe Schnittstelle zu Programmausführung und E/A-Zyklussteuerung.

Programmausführung

Die Programmausführungs-Subsysteme besitzen Echtzeit-Prozeßpriorität. Das heißt, daß die Programmausführung vor allen anderen normalen Anwendungen CPU-Zeit zugeteilt bekommt (selbst vor Mausaktualisierung und Plattenzugriff).

E/A-Zyklussteuerung

Die Subsysteme der E/A-Zyklussteuerung besitzen Echtzeit-Prozeßpriorität. Das heißt, daß die E/A-Zyklussteuerung vor allen anderen normalen Anwendungen CPU-Zeit zugeteilt bekommt (selbst vor Mausaktualisierung und Plattenzugriff).

Beim Anlauf sucht die E/A-Zyklussteuerung nach E/A-Modulen. Werden in der Rückwandplatine der Steuerung keine Module erkannt, erscheint eine Meldungsbox am Bildschirm. Drücken Sie Schaltfläche "Simulation", um die fehlende Hardware zu ignorieren und die PC Control Software im Simulationsmodus laufen zu lassen. Im Simulationsmodus stehen die Ein- und Ausgänge zwar resident in einer Speichertabelle, ihre Werte werden aber nicht von den E/A-Geräten gelesen oder dorthin geschrieben. Die Simulations-Anwendersoftware kann in das Eingangs-Speicherabbild schreiben und aus dem Ausgangs-Speicherabbild lesen, um die E/A in der Umgebung zu simulieren.

Aus der Bedienerschnittstelle heraus können Sie den Programmeditor aktivieren. Drücken Sie hierzu die entsprechende Schaltfläche oder wählen das entsprechende Menüelement aus dem Werkzeugmenü aus.

Ereignisprotokoll

Im Ereignisprotokoll werden zeitgestempelte Systemereignisse, Meldungen und Fehler gespeichert. Das Ereignisprotokoll wird von den Runtime-Subsystemen automatisch gestartet und angehalten. Das Ereignisprotokoll muß laufen, damit Meldungen zum Ausgabefenster funktionieren.

Hinweis

Um ein System zu konfigurieren, das nur aus Runtime besteht, kopieren Sie einen vollständigen Projektordner aus einem Entwicklungssystem zum Runtime (achten Sie darauf, daß beide Systeme den gleichen Pfad zu Programm und Projektdateien benutzen). Eine andere Methode besteht darin, dem System vorübergehend eine Entwicklerlizenz zu erteilen, das gewünschte Projekt zu aktivieren, und dann zu einer Runtime-Lizenz zurückzugehen. Das System, das nur aus Runtime besteht, kann nicht das Projekt wechseln.

Programmeditor

Mit dem Programmeditor können Sie Ihre Arbeit projektweise organisieren. Aus dem Programmeditor heraus können Sie eine beliebige Anzahl Projekte erstellen und verwalten.

Gleichzeitig kann jeweils nur ein Projekt geöffnet sein. Dieses Projekt wird "aktives Projekt" genannt. Mit jedem Projekt ist eine aktive Konfiguration verknüpft. Wird ein Projekt geöffnet, wird die aktive Konfiguration für dieses Projekt aktiviert. Wird ein neues Projekt geöffnet, werden alle Programme beendet.

Zugriffsebenen

Über Zugriffsebenen und Zugangscodes (Passworte) wird der Zugang zu Anwenderprogrammen, Bedienerschnittstellen-Menüs und Konfigurationsdaten gesteuert. Es gibt fünf verschiedene Zugriffsebenen, jede mit einem eigenen Zugangscode. In der nachstehenden Tabelle werden die Privilegien der einzelnen Ebenen beschrieben. Die unterste Zugriffsebene ist 0; jede höhere Ebene besitzt die Privilegien der darunterliegenden Ebenen.

Zugriffsebene	Privilegien
0	Aktivierung von Bedienermenüs und Benutzung der Bedienelemente. Ablaufplanprogramme über die Bedienerschnittstelle auswählen und laufen lassen.
1	Programme öffnen, anschauen, ausführen und anhalten.
2	Programme in Ablaufsprache, strukturiertem Text und Anweisungsliste editieren. Projektmanagementfunktionen ausführen.
3	Kontaktplanprogramme editieren. Bedienerschnittstelle modifizieren. Systemkonfigurationsdateien modifizieren.
4	Passworte für Zugriffsebenen 1 bis 4 verändern.

Der Standard-Zugangscode ist 4567.

Hinweis

Wenn Sie bei geöffnetem Programmeditor die Zugriffsebene innerhalb der Bedienerschnittstelle wechseln, ändert sich dadurch nicht die Zugriffsebene des Programmeditors.

Zugangscode eingeben

Programmeditor und Bedienerschnittstelle verlangen Zugangscodes. Der Programmeditor zeigt beim Hochlaufen automatisch das Zugriffsebenen-Tastenfeld an.

Zugangscode eingeben:

In:	müssen Sie:
dem Programmeditor	auf die Zahlen des Tastenfeldes klicken, die Ihrem Zugangscode entsprechen, und dann auf *OK* klicken.
der Bedienerschnittstelle	1. in der Menüleiste auf *Zugang* klicken und *Passwort* auswählen. Hierauf erscheint ein Tastenfeld. 2. auf die Zahlen des Tastenfeldes klicken, die Ihrem Zugangscode entsprechen, und dann auf *OK* klicken.

Zugriffsebene auf 0 einstellen:

Klicken Sie im Zugangs-Tastenfeld auf die Schaltfläche *Aufheben*.

Zugriffsebene wechseln

Der Zugangscode für die einzelnen Ebenen kann nur über die Bedienerschnittstelle mit einem Zugangscode für Ebene 4 verändert werden. Für jede Ebene gibt es einen eigenen Zugangscode.

Zugriffsebenen-Passwort verändern:

1. In der Menüleiste auf *Zugang* klicken und *Passwort eingeben* auswählen. Hierauf erscheint ein Tastenfeld.
2. Klicken Sie viermal auf die Taste '*'. Hierauf erscheint die Meldung "Geben Sie die Zugriffsebene ein, die Sie ändern wollen".
3. Klicken Sie auf die Nummer der Zugriffsebene, die Sie ändern wollen, und dann auf *OK*.
4. Klicken Sie auf die vier neuen Ziffern. Hierauf erscheint die Meldung "Neues Passwort nochmals eingeben".
5. Klicken Sie nochmals auf die vier Ziffern des Passwortes und anschließend auf *OK*. Haben Sie das Passwort richtig bestätigt, wird es geändert.

Hinweis

Klicken Sie im Tastenfeld auf AUFHEBEN, um die Zugriffsebene auf Null einzustellen.

Werden Steuerfunktionen mit Bedienelementen verknüpft (im Bedienerschnittstellen-Editiermodus), können Sie eine Zugriffsebene angeben, mit der die Benutzung der einzelnen Bedienelemente geregelt wird.

Tastaturbefehle für die Bedienerschnittstelle

Taste(n)	Aktion
<ESCAPE>	Bricht die meisten Operationen und Dialogboxen ab.
<ENTER>	Zum Editieren des ausgewählten Bedienelements.
<DELETE>	Löscht das ausgewählte Bedienelement.
<CONTROL> oder <SHIFT>	Bei der Bedienelemente-Auswahl: Fügt ein noch nicht ausgewähltes Bedienelement der ausgewählten Gruppe hinzu, oder nimmt ein ausgewähltes Bedienelement von der Gruppe der ausgewählten Bedienelemente weg.
Pfeiltaste	Bewegt das (die) ausgewählte(n) Bedienelement(e) um einen Bildpunkt in die durch die Pfeiltaste dargestellte Richtung.

Menübeschreibungen

Dateimenü

Element	Taste	Beschreibung
Neuer Editor		Erstellt eine neue Programmdatei.
Editor öffnen		Öffnet eine bestehende Programmdatei.
Schließen	-	Schließt alle mit der aktiven Programmdatei verknüpften Fenster.
Speichern		Speichert die aktive Programmdatei.
Speichern als	-	Speichert das aktive Programm unter einem anderen Namen.
Alles speichern	-	Speichert alle offenen Programmdateien und zugehörigen Projektinformationen.
Drucken...		Druckt ein Programm.
Querverweise drucken	-	Druckt Programmvariablen und wo und wie oft diese im Programm verwendet werden. Diese Option erscheint nur, wenn eine Programmdatei offen ist.
Einstellung drucken	-	Gestattet einen Druckerwechsel und ein Verändern der Druckoptionen.
Neue Konfiguration	-	Öffnet das Konfigurations-Dienstprogramm mit einer neuen E/A-Konfigurationsdatei.
Konfiguration öffnen	-	Öffnet eine bestehende E/A-Konfigurationsdatei im Konfigurations-Dienstprogramm.
Konfiguration speichern		Speichert die aktive Konfiguration.
Konfiguration speichern als		Speichert die aktive Konfiguration unter einem neuen Namen oder Pfad.
CSV in Konfiguration importieren		Importiert eine Konfigurationsdatei im CSV-Datenformat.
Konfiguration zu CSV exportieren		Exportiert die aktive Konfiguration zu einer Datei im CSV-Datenformat, auf die über einen Texteditor oder Tabellenkalkulationsprogramm zugegriffen werden kann.
Liste zuletzt benutzter Dateien	-	Listet die letzten vier vom Programmeditor aufgerufenen Dateien auf.
Exit	-	Schließt die PC CONTROL Software. Sind noch nicht gespeicherte Programme offen, werden Sie aufgefordert, diese zu speichern.

Editiermenü

Element	Taste	Beschreibung
Rückgängig		Macht die letzte Aktion rückgängig..
Wiederherstellen		Stellt die zuvor rückgängig gemachte Aktion wieder her.
Ausschneiden		Schneidet das selektierte Objekt aus und legt es in die Zwischenablage.
Kopieren		Kopiert das selektierte Objekt aus und legt es in die Zwischenablage.
Einfügen		Fügt den Inhalt der Zwischenablage ein.
Löschen	-	Löscht das ausgewählte Objekt.
Alle auswählen		Wählt den gesamten Text eines Dokuments mit strukturiertem Text oder Anweisungsliste aus.
Element editieren	-	Öffnet die Dialogbox für das selektierte Programmelement.
Neues Element	-	Fügt an dem selektierten Punkt im Programm ein KOP- oder AS-Element ein.
Neuer Funktionsblock	-	Fügt an der selektierten Stelle im Programm ein Funktionsblockelement ein. Nur verfügbar, wenn ein Kontaktplanprogramm zum Editieren geöffnet ist.
Neues AS-Element	-	Fügt ein neues AS-Element ein.
Schritteigenschaften	-	Bearbeitet einen Ablaufplan-Schritt.
ST-Anweisung einfügen	-	Fügt eine strukturierte Textanweisung ein.
ST-Funktionsaufrufe einfügen	-	Fügt eine strukturierte Textfunktion oder einen Funktionsblock ein.
AL-Anweisung einfügen		Fügt eine Anweisungslisten-Anweisung ein.
AL-Funktionsaufrufe einfügen	-	Fügt eine Anweisungslisten-Funktion oder einen Funktionsblock ein.
Suchen...		Sucht den angegebenen Text.
Nächstes suchen		Sucht nach dem nächsten Auftreten des angegebenen Textes.
Vorheriges suchen		Sucht nach dem vorherigen Auftreten des angegebenen Textes.
Ersetzen...	-	Ersetzt den angegebenen Text durch neuen Text.
Gehe zu…	-	Springt zu der angegebenen Strompfadnummer. Nur verfügbar, wenn ein Kontaktplanprogramm zum Editieren geöffnet ist.
Verfolgen...		Sucht nach dem nächsten Vorkommen einer Ausgangsspule mit dem gleichen Symbolnamen wie der selektierte Kontakt. Nur verfügbar, wenn ein Kontaktplanprogramm zum Editieren geöffnet ist.

Element	Taste	Beschreibung
Einfügemodus		Fügt neue Elemente vor dem selektierten Punkt ein.
Anhängemodus		Hängt neue Elemente hinter dem selektierten Punkt an.
Boolesche Transitionen	-	Stellt beim Editieren eines Ablaufplans die Standardtransition für alle offenen Ablaufplanprogramme auf den Booleschen Typ und nicht auf den Kontaktplantyp ein. Verändert keine bestehenden Transitionen. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Algorithmen verriegeln	-	Beim Editieren eines Ablaufplanprogramms können Sie die Schrittalgorithmen mit einem Passwortschutz versehen. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.

Anzeigemenü

Element	Taste	Beschreibung
Symbolleiste	-	Zeigt Editorbefehle als Symbole an.
Statusleiste	-	Zeigt die Statusleiste des Editorfensters an.
Kontakt-/Spulenleiste	-	Beim Bearbeiten eines Kontaktplanprogramms zeigt die Kontakt-/Spulenleiste die Programmelemente an, die Sie im Programm verwenden können. Nur verfügbar, wenn ein Kontaktplanprogramm zum Editieren geöffnet ist.
AS-Leiste	-	Beim Bearbeiten eines Ablaufplanprogramms zeigt die AS-Symbolleiste die Programmelemente an, die Sie im Programm verwenden können. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Funktionsblockpalette	-	Beim Bearbeiten eines Kontaktplanprogramms zeigt die Funktionsblockpalette die Funktionsblöcke an, die Sie im Programm verwenden können. Nur verfügbar, wenn ein Kontaktplanprogramm zum Editieren geöffnet ist.
Anwendungssymbolpalette		Beim Bearbeiten eines Ablaufplanprogramms zeigt die Anwendungssymbolpalette die Symbole an, die Sie im Programm verwenden können. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Zubehörleiste		Zeigt die Zubehörleiste von strukturiertem Text oder Anweisungsliste an, die automatisch Programmanweisungen einfügt.
Variablen überwachen/fixieren		Zur Laufzeit zeigt dieses Fenster Variablen und deren aktuelle Werte an. Sie können zum Austesten Ihres Programms für diese Variablen neue Werte angeben (fixieren). Nur verfügbar, wenn ein Programm zum Editieren geöffnet ist.
E/A-Fehler		Zeigt das E/A-Fehlerfenster an.
Ausgabefenster	-	In diesem Fenster erscheinen Meldungen, die zur Laufzeit auftreten.
Programmstatus	-	Öffnet das Programmstatusfenster.

Element	Taste	Beschreibung
Parserfehler anzeigen		Zeigt das Fenster für Parserfehler an.
Programmkommentare		Zeigt oder versteckt alle im Programm eingegebenen Programmkommentare. Nur verfügbar, wenn ein Programm zum Editieren geöffnet ist.
Auf Fenstergröße skalieren		Stellt die Programmgröße so ein, daß es in das aktuell angezeigte Fenster paßt. Nur verfügbar, wenn ein Programm zum Editieren geöffnet ist.
KOP-Warnungen	-	Schaltet die Hinweise durch KOP-Warnungen ein bzw. aus. Die Warnungen treten während der Syntaxanalyse eines Programms auf.
Symbolort anzeigen		Zeigt für E/A-Symbole Chassis/Steckplatz/Punkt an.
FB-Details umschalten		Schaltet die Anzeige von Symbolnamen und Echtzeitdaten und Kontaktplan-Funktionsblöcken ein/aus. Nur verfügbar, wenn ein Kontaktplanprogramm zum Editieren geöffnet ist.
Alle Schritte	-	Beim Bearbeiten eines Ablaufplanprogramms können Sie mit dieser Option angeben, daß Schritte mit ihren Namen, Beschreibungen, zugehörigen Symbolen oder ihrem Programmcode angezeigt werden. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Symbolaufzählungen anzeigen	.m	Schaltet die Symbolaufzählungen ein, die in der Dialogbox "Systemoptionen" eingestellt wurden.

Projektmenü

Element	Beschreibung
Neu	Erstellt ein neues Projekt zum Speichern verwandter Anwenderprogramme und Konfigurationen.
Öffnen	Öffnet ein bestehendes Projekt.
Kopieren	Kopiert ein bestehendes Projekt (einschließlich aller Anwenderprogramme und Konfigurationsdateien) zu einem neuen Namen.
Umbenennen	Umbenennen eines bestehenden Projektes. Hat keine Auswirkungen auf die Anwenderprogramme oder Konfigurationsdateien in dem Projekt.
Konfiguration aktivieren	Aktiviert die aktuelle Konfigurationsdatei. Die aktuelle Konfigurationsdatei wird in der Statusleiste unten am Bildschirm angezeigt.

Ausführungsmenü

Element	Beschreibung
Syntaxanalyse durchführen	Kompiliert das Programm
Lauf	Kompiliert das Programm und führt es anschließend aus.
RUN mit Austesten	Führt das Programm aus, bis eine strukturierte Textanweisung BREAK erreicht wird, stoppt dann die Programmausführung und versetzt das Programm in den Status "Break". BREAK-Anweisungen werden ignoriert, wenn ein RUN-Befehl gegeben wird. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Einzelschritt	Führt jeweils ein einzelnes Kommando aus. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Lauf mit Neustart	Markiert das Programm, daß es bei jedem Laufzeitstart mit der Ausführung beginnt. Ein abgebrochenes oder fehlerhaftes Programm wird nicht mehr mit Lauf mit Neustart markiert.
Stop	Stoppt die Programmausführung. Wird das Programm wieder gestartet, beginnt es mit der Ausführung an dem Punkt, an dem es angehalten wurde. Nur verfügbar, wenn ein Ablaufplanprogramm zum Editieren geöffnet ist.
Abbruch	Hält die Programmausführung an. Wird das Programm wieder gestartet, beginnt es mit der Ausführung am Programmanfang.
Nothalt rücksetzen und E/A-Fehler löschen	Setzt die E/A-Treiber zurück und stellt die Datenverbindung zur E/A-Hardware wieder her.
Runtime-Subsysteme starten	Startet die Runtime-Subsystemprogramme.

Symbolmenü

Element	Beschreibung
Neues Anwendungssymbol	Aufruf der Anwendungssymbolschrittbibliothek , wodurch Sie vordefinierte Schritte in der Bibliothek erstellen können.
Anwendungssymbol editieren	Bearbeiten vordefinierter Schritte in der Anwendungssymbolbibliothek.
Anwendungssymbol löschen	Löscht vordefinierte Schritte aus der Anwendungssymbolbibliothek.

Werkzeugmenü

Element	Taste	Beschreibung
Bedienerschnittstelle	-	Startet die integrierte Bedienerschnittstelle
Symbolmanager		Ruft den Symbolmanager auf, mit dem Sie Variablen erstellen und bearbeiten können.
Aktionsmanager	-	Aufruf des Aktionsmanagers, mit dem Sie AS-Aktionen umbenennen und löschen können.
Kontaktplantransitionsmanager	-	Aufruf des Transitionsmanagers, mit dem Sie AS- und KOP-Transitionen umbenennen und löschen können.
Symbole exportieren	-	Exportiert alle globalen Symbole zu einer *.SNF-Datei. Nur mit der CIMPLICITY-Bedienerschnittstelle verfügbar.
Systemoptionen	-	Öffnet eine Dialogbox, in der Sie die Größe des Heap-Speicher und der Speichervariablen Datentabelle, die Anzeige und die Symbolaufzählungs-Vorzugswerte einstellen können.

Lizenzmenü

Element	Beschreibung
Autorisierung	Mit dieser Option können Sie eine Software "Lizenz" für die PC CONTROL Software erstellen. Wenn Sie GE Fanuc Automation Ihren "Anlagencode" per Telefon oder Fax mitteilen, erhalten Sie einen "Anlagenschlüssel" zur Autorisierung Ihrer Software. Diese Option ist nur aktiviert, wenn die Software im Demonstrationsmodus gestartet wurde.
Hochrüst-Autorisierung	Wenn Sie die Funktionalität von PC CONTROL erweitern wollen, können Sie mit der Option Hochrüst-Autorisierung einen neuen "Anlagencode" erzeugen. Diese Option wird erst freigegeben, nachdem die Software autorisiert wurde.
Registertransfer	Über diese Option können Sie eine Softwarelizenz von einem Computer zu einem anderen übertragen. Legen Sie eine leere Floppy Disk in einen nicht autorisierten Computer ein und wählen dann "Registertransfer", um für den Computer eine eindeutige Signaturdatei zu schreiben. Diese Option ist nur aktiv, wenn die Software im Demonstrationsmodus gestartet wurde.
Austransfer	Austransfer ist der zweite Schritt bei der Übertragung einer Softwarelizenz von einem Computer zu einem anderen. Nachdem Sie den Registertransfer im nicht autorisierten Computer beendet haben, legen Sie die Floppy Disk in den autorisierten Computer ein und wählen die Option "Austransfer". Die Softwarelizenz wird vom Computer entfernt und auf die Floppy Disk geschrieben. Diese Softwarelizenz kann nur zu dem Computer übertragen werden, die die ursprüngliche Signatur auf die Floppy Disk geschrieben hat. Diese Option ist nur in einem autorisierten Computer aktiv.

Element	Beschreibung
Eintransfer	Eintransfer ist der letzte Schritt bei der Übertragung einer Softwarelizenz von einem Computer zu einem anderen. Legen Sie die Floppy Disk in den Computer ein, zu dem die Softwarelizenz übertragen werden soll. Wählen Sie "Eintransfer", um die Softwarelizenz zu dem Computer zu übertragen. Diese Softwarelizenz kann nur zu dem Computer übertragen werden, die die ursprüngliche Signatur auf die Floppy Disk geschrieben hat.
Temporäre Autorisierung	Mit der temporären Autorisierung können Sie temporär eine volle Entwicklungsautorisierung in einem Laufzeitsystem erhalten. Jedes Laufzeitsystem wird mit 30 Minuten kostenloser temporärer Autorisierung ausgeliefert. Weitere Zeit kann dazugekauft werden. Gültig für Systeme, die nur mit Runtime ausgestattet sind. Eine kostenlose Kopie des Dienstprogramms "Temporäre Autorisierung" erhalten Sie über die in der Autorisierungs-Dialogbox angegebene Telefonnummer. Wenn Sie temporäre Autorisierung wählen, erscheinen die Schaltflächen *Temporäre Autorisierung* und die Restzeit der temporären Autorisierungszeit wird angezeigt: *OK* - aktiviert die temporäre Autorisierung und startet den temporären Autorisierungstimer (solange noch Zeit übrig ist). *Abbrechen* - Rückkehr zum Laufzeitsystem. *Weitere Minuten hinzufügen* - hiermit können Sie für dieses System zusätzliche temporäre Autorisierung erwerben. Rufen Sie die in der Autorisierungs-Dialogbox angegebene Telefonnummer an und geben Sie Ihren Code aus dem Feld *Code* an. Sie erhalten dann einen Schlüssel, den Sie im Feld *Schlüssel* eingeben müssen, um die zusätzliche Zeit der temporären Autorisierung hinzuzufügen.
Ursprüngliche Autorisierung rücksetzen	Schaltet die temporäre Autorisierung ab und stoppt den temporären Autorisierungstimer. Gültig für Systeme, die nur mit Runtime ausgestattet sind.
Notfallautorisierung	Liefert eine volle Autorisierung aller Eigenschaften für die Dauer von vier Tagen. Nach vier Tagen kehrt PC CONTROL in den Demonstrationsmodus zurück. Nach der Anwahl werden Sie gewarnt, daß dies bis zur nächsten Neuformatierung der Festplatte eine einmalige Autorisierung ist. Fahren Sie fort, dann wird auf die Festplatte eine Signatur eingetragen.

Symbolleisten und Statusleiste

Die Symbolleisten enthalten Schaltflächen für häufig ausgeführte Funktionen.

Standard-Symbolleiste

Schaltfl äche	Werkzeug
	Neu
	Öffnen
	Speichern
	Rückgängig
	Wiederherstellen
	Ausschneiden
	Kopieren
	Einfügen
	Vorher einfügen
	Danach einfügen
	Drucken
	Über
	Kontexthilfe
	Kommentaranzeige
	Auf Fenstergröße skalieren
	Symbolmanager
	Überwachungsfenster
	Boolescher Transitionsmodus

Kontaktplan-Symbolleiste

Schaltfläche	Werkzeug
	Kontakt
	Strompfadetikett
	Spulenwerkzeug
	Sprungmerker-Werkzeug
	Transitionsmerker-Werkzeug
	Verzweigungswerkzeug
	Neuer Strompfad

AS-Symbolleiste

Schaltfläche	Werkzeug
	Auswahlwerkzeug
	Schritt
	Makroschritt
	Aktion
	Transition
	Marke
	Sprung
	Schleife
	Auswahlverzweigung
	Simultanverzweigung
	Anwendungssymbol
	Kommentar hinzufügen

Statusleiste

Die Statusleiste befindet sich unten am Programmeditor-Fenster. Die Statusleiste zeigt den Namen der aktuellen Konfiguration und den Status von

- NUM Lock
- SCROLL Lock
- CAPS Lock
- Tastaturtasten
- Einfüge- und Anhängemodus

Antworten auf häufig gestellte Fragen

Benötigt PC Control eine SPS ?

- PC Control benötigt für die Logikbearbeitung keine SPS.
- PC Control führt die Logikbearbeitung im PC durch, ohne dafür eine SPS zu benötigen.
- PC Control kann zur Steuerung von E/A-Chassis, Modulen und anderen Geräten im SPS-Stil Scannermodule im PC verwenden.

Führt PC Control eine Paritätsprüfung durch ?

PCs können so erworben und konfiguriert werden, daß sie im gesamten Systemspeicher eine Paritätsprüfung durchführen.

PC Control führt bei Anwenderprogrammen keine Paritätsprüfung durch, berechnet jedoch bei allen laufenden Anwenderprogrammen im Hintergrund die Prüfsummen. Wird ein Fehler erkannt, dann wird das Anwenderprogramm angehalten, als fehlerhaft markiert und der Bediener kann bestimmen, was zur Fehlerbehebung geschehen soll.

Was geschieht bei einem Paritätsfehler ?

Wird im System ein Paritätsfehler festgestellt, hält Windows NT an und gibt ein Fehlermenü mit einem blauen Hintergrund aus, das einen Datenbusfehler sowie die Adresse, bei der der Fehler erkannt wurde, anzeigt. Zum Fortsetzen muß das System neu gebootet werden.

Das PC Control System verwendet Überwachungszeitgeber und die in einigen E/A-Systemen vorhandenen Eigenschaften "sicherer E/A-Zustand", um einen Ausfall der Aktivitäten der PC Control Steuerung zu erkennen und die Ausgänge in einen vordefinierten sicheren Zustand zu versetzen.

Weitere Informationen können Sie von GE Fanuc Automation erhalten.

Was geschieht bei einem Spannungsausfall ?

PC und E/A-System gehen in den Spannungsausfallzustand. Bei Spannungsrückkehr nimmt das E/A-System den Zustand an, den es normalerweise oder per Konfiguration annimmt, wenn Versorgungsspannung und Kontrolle ausgefallen sind. Enthält das System nichtflüchtige oder batteriegepufferte Speicher, dann kann eine PC CONTROL Anwendung E/A und Anwenderprogramme ab dem Unterbrechungspunkt wieder starten.

Kann die PC Control Software automatisch gestartet werden ?

Windows NT kann so konfiguriert werden, daß die PC Control Software und das Anwenderprogramm automatisch beim Booten gestartet werden.

Weitere Informationen zu diesem Thema finden Sie unter "Programme für automatischen Ablauf konfigurieren” in Kapitel 5

Funktioniert die PC Control Software mit einer unterbrechungsfreien Stromversorgung ?

Windows NT kann für den Betrieb mit einer unterbrechungsfreien Stromversorgung (USV) konfiguriert werden, die bei Ausfall der Netzspannung das Gerät mit Spannung versorgt. Windows NT wird informiert, wenn die Netzspannungsversorgung ausfällt, und kann zu entsprechenden Gegenmaßnahmen veranlaßt werden.

PC CONTROL kann auch die von der USV empfangenen Signale "Spannungsausfall" und "Batteriespannung niedrig" dazu verwenden, anwenderprogrammierte Aktionen durchzuführen. Weitere Informationen finden Sie unter "USV-Konfiguration” auf Seite 2-33.

Abschnitt 2: Mit Projekten und Anwendungen arbeiten

Projekte verwalten

Über das Projektmanagement kann die Arbeit in kleine logische Abschnitte, die Projekte, aufgeteilt werden. Jedes Projekt enthält eine Gruppe aus Programm, Konfiguration und Bedienerschnittstellendateien, die gemeinsam verwendet werden.

Mit dem Programmeditor können Anwender-Steuerungsprogramme in KOP, AS, strukturiertem Text und Anweisungsliste editiert werden. Alle Projektmanagementfunktionen werden vom Programmeditor durchgeführt. Die folgenden Hinweise beziehen sich auf Projekte:

- Der Programmeditor kann jeweils nur ein Projekt geöffnet haben.
- Der Projektname sollte den allgemeinen Zweck der Dateien kennzeichnen.
- Das aktive Projekt steht in der Kopfzeile des Programmeditor-Fensters.

Ein neues Projekt erstellen

1. Starten Sie den Programmeditor.
2. Klicken Sie in der Menüleiste auf *Projekt* und wählen Sie dann *Neu*. Hierauf erscheint die Dialogbox "Neues Projekt".
3. Geben Sie den Namen des neuen Projektes ein und klicken dann auf *OK*.

Hinweis

War zuvor ein aktives Projekt offen, ist es jetzt geschlossen und das neue Projekt wird zum aktiven Projekt.

Ein Projekt öffnen

1. Starten Sie den Programmeditor.
2. Klicken Sie in der Menüleiste auf *Projekt* und wählen Sie dann *Öffnen*. Hierauf erscheint die Dialogbox "Projekt öffnen".
3. Klicken Sie zunächst auf das Projekt, das Sie öffnen wollen, und dann auf *OK*.

Wird ein Projekt geöffnet, dann wird es zum aktiven Projekt. Das zuvor aktive Projekt wird zusammen mit allen anderen offenen Programmdateien geschlossen. Wird ein Projekt geöffnet, dann werden auch alle Programmdateien wieder geöffnet, die zuvor in dem Projekt geöffnet waren. Außerdem wird die Konfigurationsdatei aktiviert, die in dem aktivierten Projekt zuletzt aktiv war.

Ein Projekt kopieren

1. Starten Sie den Programmeditor.
2. Klicken Sie in der Menüleiste auf *Projekt* und wählen Sie dann *Kopieren*. Hierauf erscheint die Dialogbox "Projekt kopieren von".
3. Klicken Sie zuerst auf das Projekt, das Sie kopieren wollen, und dann auf *OK*. Hierauf erscheint die Dialogbox "Projekt kopieren nach".
4. Geben Sie den Namen der Kopie ein und klicken dann auf *OK*.

Beim Kopieren eines Projektes werden das gesamte Programm, die Bedienerschnittstelle und die Konfigurationsdateien des Projekts in das neue Projekt kopiert und das neue Projekt wird zum aktiven Projekt. Programmdateien, die im Quellprojekt zuvor offen waren, werden in das neue Projekt kopiert und geöffnet. Die im Quellprojekt aktive Konfigurationsdatei wird in das neue Projekt kopiert und aktiviert.

Ein Projekt umbenennen

1. Starten Sie den Programmeditor.
2. Klicken Sie in der Menüleiste auf *Projekt* und wählen Sie dann *Umbenennen*. Hierauf erscheint die Dialogbox "Projekt umbenennen von".
3. Klicken Sie zunächst auf das Projekt, das Sie umbenennen wollen, und dann auf *OK*. Hierauf erscheint die Dialogbox "Projekt umbenennen in".
4. Geben Sie den neuen Namen des Projektes ein und klicken dann auf *OK*.

Nachdem ein Projekt umbenannt wurde, wird es zum aktiven Projekt.

Eine Konfiguration aktivieren

1. Wählen Sie im Menü "Projekt" die Option "Aktivieren". Hierauf erscheint die Dialogbox "Globale Konfigurationsdatei auswählen". Sie zeigt eine Liste der für das aktuelle Projekt definierten Konfigurationsdateien an.

2. Wählen Sie die Konfiguration, die Sie aktivieren wollen, und klicken dann auf OK.

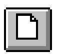
Mit Anwenderprogrammen arbeiten

Mit der Programmverwaltung können Sie Programme in Kontaktplanlogik (KOP), Ablaufsprache (AS), strukturiertem Text (ST) und Anweisungsliste (AL) erstellen und speichern. Sie können gleichzeitig viele Programme öffnen. Mit dem Fenstermenü können Sie zwischen offenen Programmen umschalten. Sie können auch mit der Maus auf ein teilweise sichtbares Programm klicken und es dadurch nach oben zur Anzeige bringen. Die aktiven Programme werden in der Kopfzeile des Programmeditors angezeigt.

Ein neues Programm erstellen

Ein neues Programm in einem neuen Fenster erstellen:

1. Klicken Sie auf *Datei* und wählen dann *Neu* oder klicken Sie auf die Schaltfläche "neues Programm" in der Editor-Symbolleiste.



Es erscheint eine Dialogbox, in der Sie ein neues Kontaktplanprogramm (KOP), einen Ablaufplan (AS), ein Dokument in strukturiertem Text, oder eine Anweisungsliste spezifizieren können.

2. Wählen Sie den Programmtyp, den Sie erstellen wollen, und klicken dann auf *OK*. Hierauf wird ein neues Editorfenster des entsprechenden Typs geöffnet. Das Programm erhält einen Standardnamen (zum Beispiel *KOP1*).

Ein Programm öffnen

Um ein bestehendes Programm in einem neuen Fenster zu öffnen:

Klicken Sie auf *Datei* und wählen dann *Öffnen* oder klicken Sie auf die Schaltfläche "Programm öffnen" in der Editor-Symbolleiste.



Sie können gleichzeitig viele Programme öffnen. Mit dem Fenstermenü können Sie zwischen offenen Programmen umschalten. Sie können auch mit der Maus auf ein teilweise sichtbares Programm klicken und es dadurch nach oben zur Anzeige bringen. Ist kein Projekt aktiv, wenn der Befehl "Datei öffnen" ausgeführt wird, dann erscheint die Dialogbox "neues Projekt" oder "Projekt öffnen".

Eine Datei speichern

Das aktive Programm unter aktuellen Namen und Verzeichnis speichern:

Klicken Sie auf *Datei* und wählen dann *Speichern* oder klicken Sie auf die Schaltfläche "Programm speichern" in der Editor-Symbolleiste.



Wenn Sie ein Programm zum ersten Mal speichern, dann zeigt der Programmeditor die Dialogbox "Speichern als" an, so daß Sie Ihrem Programm einen Namen geben können. Wollen Sie beim aktiven Programm vor dem Speichern den Namen oder das Verzeichnis ändern, dann verwenden Sie den Dateibefehl *Speichern als*.

Datei unter neuem Namen speichern

Klicken Sie auf *Datei* und wählen Sie dann *Speichern als*. Der *Programmeditor* zeigt die Dialogbox "Speichern als" an, in der Sie den Namen des Programms ändern können.

Ein Programm drucken

Drucken eines Programms in AS, KOP, ST oder AL:

Klicken Sie auf *Datei* und wählen dann *Drucken* oder klicken Sie auf die Schaltfläche "Programm drucken" in der Editor-Symbolleiste.



Um die Druckereinstellung zu verändern, klicken Sie auf *Datei* und wählen dann den Menübefehl *Druckeinstellung*. Oder drücken Sie die Schaltfläche *Einstellung* in der Druck-Dialogbox.

Beim Ausdrucken eines Programms in Ablaufsprache (AS) wird zuerst der Ablaufplan gedruckt. Im Ablaufplan eingebettete Aktionen werden im Anschluß an den Ablaufplan in alphabetischer Reihenfolge gedruckt. Nach den eingebettenen Aktionen werden in alphabetischer Reihenfolge alle eingebetteten Kontaktplanlogik-Transitionengedruckt.

Programm-Querverweise drucken

Diese Funktion druckt die Verwendung der Symbole in einem Programm aus. Sie listet alle Symbole in einer Konfiguration auf und gibt an, wo und wie oft jedes Symbol im Programm verwendet wurde. Bei einem Kontaktplanprogramm werden zum Beispiel Strompfadnummer und Kontakt- oder Spulentyp angegeben.

Programm-Querverweis drucken

- Wählen Sie im Menü *Datei* die Option *Querverweise drucken*.

Anwenderprogramme verwalten

Mit der Programmverwaltung können Sie Kontaktplanprogramme (KOP) und Programme in Ablaufsprache (AS) erstellen und abspeichern. Sie können gleichzeitig viele Programme öffnen. Das aktive Programm wird in der Kopfzeile des Programmeditors angezeigt.

Zugriff auf andere offene Programme im Programmeditor:

1. Klicken Sie auf "Fenster" in der Menüleiste.
2. Klicken Sie auf das Programm, das Sie anzeigen wollen

Hinweis

Sie können auch mit der Maus auf ein nur teilweise sichtbares Programm klicken und es dadurch nach vorne bringen.

Befehle, die aus den Menüs heraus oder durch Betätigen von Schaltflächen in Symbolleisten ausgeführt werden, wirken sich immer auf das aktive Programm aus.

Anwenderprogramme schließen

Um ein Programm zu schließen:

Klicken Sie in der Menüleiste auf *Datei* und wählen Sie dann *Schließen*.

oder

Klicken Sie in der echten oberen Ecke des Fensters auf die Schaltfläche *Schließen* ..

Programme anzeigen

Wird ein Programm geöffnet, dann wird der Programminhalt in einem Fenster angezeigt. Für das gleiche Programm (in KOP, AS, AL oder ST) können Sie gleichzeitig mehrere Fenster öffnen.

Zugriff auf andere offene Programme im Programmeditor

- Wählen Sie das Programm aus der Liste unter dem Menü *Fenster* aus.
- Klicken Sie mit der Maus auf ein nur teilweise sichtbares Programm und bringen es dadurch nach vorne.

Programm auf Fenstergröße anpassen

Klicken Sie in der Menüleiste auf *Anzeigen* und wählen Sie dann die Option *Auf Fenstergröße skalieren*.

oder

Klicken Sie in der Symbolleiste auf die Schaltfläche "Auf Fenstergröße skalieren", um die Ansicht umzuschalten.



Die Funktion "Auf Fenstergröße skalieren" kann auch dazu benutzt werden, in einem Ablaufprogramm eingebettete Aktions- oder Transitions-Kontaktpläne zu skalieren.

Navigieren mit "Auf Fenstergröße skalieren"

1. Skalieren Sie das Programm so, daß es in das Fenster paßt.
2. Selektieren Sie das gewünschte Element.
3. Entfernen Sie die Fensterskalierung. Sofern es nicht in der letzten unskalierten Anzeige ist, wird das selektierte Element in die Bildschirmmitte verschoben.

Kommentare ein- und ausschalten

Sie können einem Anwenderprogramm beschreibende Kommentare hinzufügen, um die Programmfunktionen zu dokumentieren. Die Betriebsart "Kommentare anzeigen" wirkt auf alle offenen Programme.

Umschalten der Betriebsart "Kommentare anzeigen":

Klicken Sie in der Menüleiste auf *Anzeigen* und wählen *Programmkommentare*.

oder

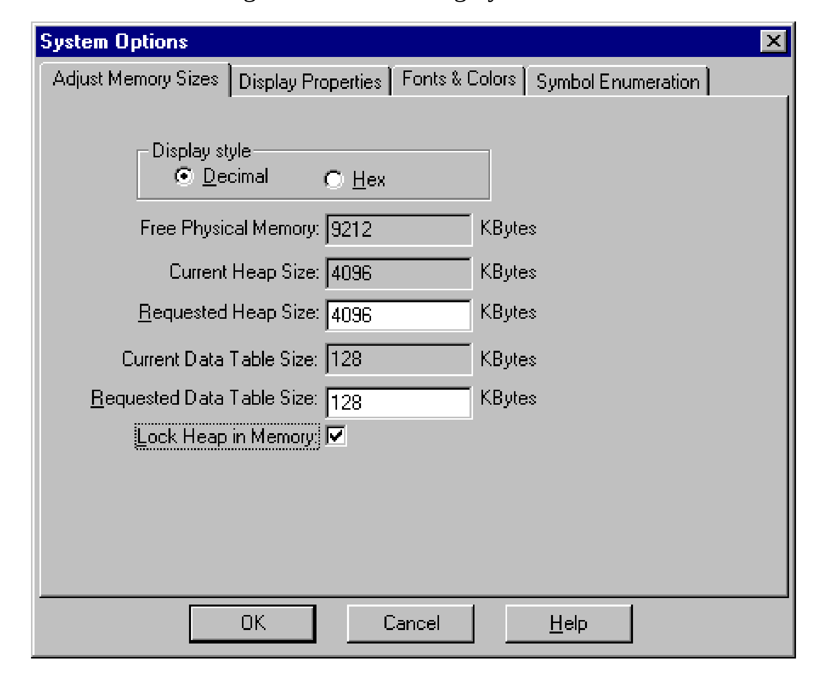
Schalten Sie die Betriebsart "Kommentare anzeigen" über die Schaltfläche "Kommentare anzeigen" in der Symbolleiste um.

Hinweis

Ein Häkchen neben "Programmelemente" und eine gedrückte Schaltfläche "Kommentare anzeigen" zeigen an, daß Kommentare sichtbar sind.

Systemoptionen einstellen

In der Dialogbox "Systemoptionen" können Sie Parameter einstellen, die das Verhalten und Erscheinungsbild des Steuerungssystems beeinflussen.



Karteikarte "Speichergrößen einstellen"

Feld	Beschreibung
Anzeigeform	Zeigt die Speichergröße als Dezimalwert oder Hexadezimalwert an.
Freier physikalischer Speicher	Der aktuell freie physikalische Speicher im Computer. Dieser Wert kann nicht verändert werden. Er ist abhängig von Art und Anzahl der offenen Windows-Anwendungen.
Aktuelle Heapgröße	Dieser Wert gibt die aktuelle Heapeinstellung an. Er kann nicht verändert werden. Der Heap ist der Speicher, der (beim Hochlauf) dem Steuerungssystem zugewiesen wird. Der Heap wird von offenen Anwenderprogrammen und Symbolen verbraucht. Ist er verbraucht, muß diese Einstellung erhöht werden. Je größer der Heap, desto kleiner wird allerdings der für andere Windows-Anwendungen verfügbare Speicher.
Geforderte Heapgröße	Hier können Sie die Heapgröße verändern. Da der Heap virtuellen Speicher von der Auslagerungsdatei verwendet, kann er größer als der physikalische Speicher sein. Reicht der Steuerungssystemspeicher während einer Editiersitzung nicht aus, dann wird Ihnen in einer Meldung vorgeschlagen, die Größe des Heapspeichers zu erhöhen.

Feld	Beschreibung
Aktuelle Datentabellengröße	Dieser Speicher wird für Symbole verwendet. Seine Größe ist abhängig von dem Symboldatentyp. Große Felder, zum Beispiel, belegen mehr Datentabellenspeicher. Reicht der Datentabellenspeicher während einer Editiersitzung nicht aus, dann wird Ihnen in einer Meldung vorgeschlagen, die Größe der Datentabelle zu erhöhen.
Geforderte Datentabellengröße	Falls erforderlich, geben Sie hier die neue Datentabellengröße ein.
Heap im Speicher einklinken	Wird dieses Feld aktiviert, verbessert sich das deterministische Verhalten des Steuerungssystems. Beim Starten des Steuerungssystems versucht es, den Heap im physikalischen RAM einzuklinken. Schlägt dies fehl, kann er virtuellen Speicher benutzen. Erfolg oder Mißerfolg dieses Versuchs werden Ihnen nicht mitgeteilt. Sie starten daher am besten das Runtimesystem des Steuerungssystems vor allen anderen Anwendungen, so daß für den Heap soviel physikalischer Speicher wie möglich vorhanden ist.

Karteikarte "Eigenschaften anzeigen"

Feld	Beschreibung
Runtime Tablett-Symbol	Ist diese Option aktiv, wird das Runtimesymbol im Tablett angezeigt; andernfalls wird es auch in der Taskleiste angezeigt. Wird es in der Taskleiste angezeigt, dann können Sie es über die Tastatur aufrufen. Wird es nur im Tablett angezeigt, müssen Sie es mit der Maus aufrufen.
Anzahl Dezimalstellen	Hierdurch wird die Anzahl der bei den Funktionsblockdetails in einem Kontaktplanprogramm angezeigten Dezimalstellen festgelegt. Je mehr Dezimalstellen Sie anzeigen, desto mehr Platz beanspruchen Sie.
Anzahl Funktionsblock-Symbolzeichen	Hier legen Sie den Platz fest, der bei den Funktionsblockdetails in einem Kontaktplanprogramm für die Anzeige von Zeichen verfügbar ist (nicht unbedingt die aktuelle Anzahl der Zeichen). Es wird eine Standard-Zeichengröße eines Großbuchstabens *X* angenommen. Benötigt ein Zeichen weniger Platz (z.B. ein kleines *I*), dann erscheinen mehr Zeichen.
Aktualisierung Editoranzeige	Dieser Wert legt fest, wie oft der Programmeditor aktualisiert wird (z.B. die Werte im Überwachungsfenster).
Anzeige im IEC-Stil	Ist dieses Feld gesetzt, werden die E/A-Punkte in der Syntax nach IEC 1131-3 angezeigt, nicht durch Symbolnamen.

Karteikarte Schriftarten und Farben

Feld	Beschreibung
Schriftarten	Legt die Schriftarten in den Programmeditoren fest.
Farben	Einstellung der verschiedenen Editorfarben.
Auf Standardfarben rücksetzen	Setzt alle Farben auf die voreingestellten Werte zurück.

Karteikarte "Symbolaufzählung"

Aufzählungen beziehen sich auf die Elemente eines komplexen Symbols. Dies ist ein Symbol mit einzeln zugänglichen Elementen: Strukturen, Funktionsblöcke usw. Die nachstehenden Kontrollkästchen arbeiten mit dem Befehl *Symbolaufzählungen anzeigen* aus dem Editormenü *Anzeigen*. Ist das zugehörige Kontrollkästchen aktiviert und *Symbolaufzählungen anzeigen* freigegeben, werden die entsprechenden Aufzählungen in den Symbollisten sichtbar. Ist das Kontrollkästchen nicht aktiviert oder *Symbolaufzählungen anzeigen* gesperrt, dann sind die entsprechenden Aufzählungen in den Symbollisten nicht sichtbar.

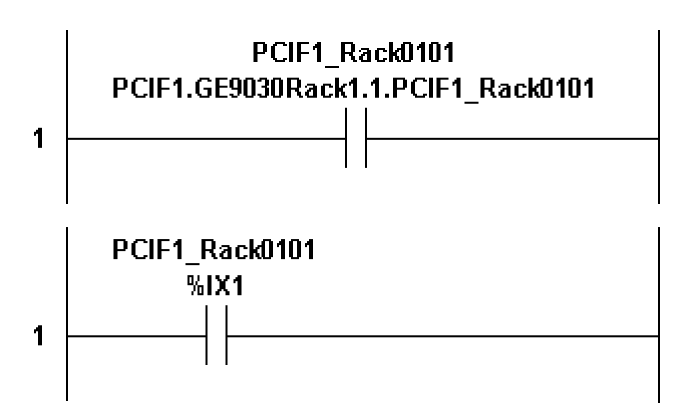
Die Anzeige der Symbolaufzählung (abhängig von der Anzahl Symbole) kann die Leistung bei der Verwendung des Editors beeinträchtigen, da in die Symbollisten mehr Symbole geladen werden müssen.

Hinweis: Wenn Sie ein Symbol vom Symbolmanager herunterziehen und ablegen erscheint selbst bei abgeschalteter Aufzählung ein Listenfeld, aus dem Sie die Aufzählung auswählen können, die Sie verwenden wollen,

Feld	Beschreibung
Funktionsblock-Aufzählungen anzeigen	Zeigt die Ein- und Ausgänge einer Funktionsblock-Instanz an.
Anwendertyp-Aufzählungen anzeigen	Zeigt die Elemente in einer Anwendertyp-Instanz an.
Achsenaufzählungen anzeigen	Zeigt die Symbole der Achsenelemente an.
Achsgruppen-Aufzählungen anzeigen	Zeigt die Symbole der Achsgruppenelemente an.
Systemobjekt-Aufzählungen anzeigen	Zeigt die mit den Instanzen von Timern (TMR), PIDs und PRGCBs verknüpften Variablen an.

Adreßanzeige im IEC-Stil

Die nachstehende Abbildung zeigt eine Adreßangabe für E/A-Punkte in der normalen Schreibweise und im Stil der IEC 1131-3. (*Symboladressen anzeigen* muß freigegeben sein). Anstelle der Symboladresse erscheint **Speicher**, wenn ein Symbol im Speicherabbild variabel ist.



Die Adreßsyntax der IEC 1131-3 ist wie folgt:

%	Direkt dargestellte Variable
I	Eingangspunkt
Q	Ausgangspunkt
X	Boolesche Variable
B	BYTE
W	WORD
D	DWORD

USV-Konfiguration

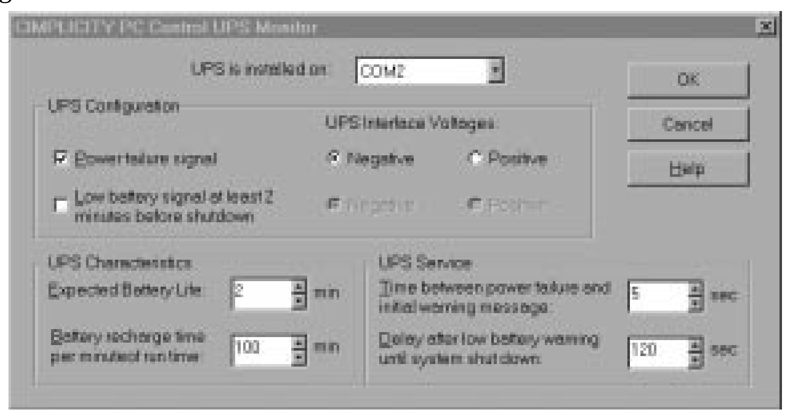
Das Steuerungssystem kann mit einer USV betrieben werden. Zur Vorbereitung einer ordnungsgemäßen Abschaltung kann das Steuerungssystem von der USV Signale empfangen. Diese Signale erscheinen dann im Symbolmanager als:

RT_POWER_FAIL	Spannungsausfall erkannt.
RT_LOW_BATTERY	Niedrige Batteriespannung erkannt.

Verwendung der USV

- Rufen Sie im Windowsmenü *Start* das Menü *PC Control Anwendungen* auf und wählen dort *USV Monitor*.

Im Tablett erscheint das Symbol für den USV-Monitor . Doppelklicken Sie auf das Symbol, wenn Sie die USV konfigurieren wollen. Hierauf erscheint die Dialogbox *PC Control USV-Monitor*.



Element	Beschreibung
USV angeschlossen an:	Geben Sie hier den seriellen Port an, an dem die USV angeschlossen ist.
USV-Konfiguration	Schlagen Sie zum Ausfüllen dieses Feldes in Ihrer USV-Dokumentation nach.
Spannungsausfallsignal USV-Schnittstellenspannung	Aktivieren Sie diese Option, wenn die USV ein Spannungsausfallsignal liefert. Ist die Option aktiviert, wählen Sie aus, ob die Signalspannung positIv oder negativ ist.
Signal für niedrige Batteriespannung USV-Schnittstellenspannung	Aktivieren Sie diese Option, wenn die USV ein Signal für niedrige Batteriespannung liefert. Ist die Option aktiviert, wählen Sie, ob die Signalspannung positIv oder negativ ist.
USV-Eigenschaften	Liefert die USV kein Signal für niedrige Batteriespannung, können Sie die USV-Eigenschaften aus den USV-Unterlagen hier eintragen.
USV-Dienst	Stellen Sie diese Werte entsprechend den Anforderungen Ihres Systems ein.

Abschalten der USV

- Klicken Sie mit der rechten Maustaste auf das Symbol "USV-Monitor" und wählen Sie die Option *Speicher-USV abschalten*.

Variablen sichern und wiederherstellen

Beim Auftreten des zugehörigen Ereignisses setzt der USV-Monitor die Variable RT_POWER_FAIL (Spannungsausfall) bzw. RT_LOW_BATTERY (niedrige Batteriespannung). Sie können über Anwenderlogik Werte in einer Datei auf der Festplatte abspeichern und dann beim Neustart diese Werte zur Wiederherstellung lesen.

Logikbeispiel zum Speichern von Variablen:

```
IF RT_LOW_BATTERY THEN


    savedata.int1:= ImportantValue;

    (* für alle zu speichernden Informationen wiederholen *)


    OPENFILE(FCBVar, FILE:= "Backup.dat");

    WRITEFILE(FCBVar, IN:= savedata );

    CLOSEFILE(FCBVar);


END_IF;
```

Abschnitt 3: Arbeitsweise des Programms - Übersicht

Konfiguration aktivieren

Beim Aktivieren einer Konfiguration werden folgende Vorgänge ausgeführt:

- Der Globalspeicher wird initialisiert
- Wurden die Ausgänge als gesperrt konfiguriert, werden die Schnittstelle freigegeben, die Ausgänge gesperrt (fixieren funktioniert nicht) und der Programmzylus gestartet.
- Wurden die Ausgänge als freigegeben konfiguriert, werden der voreingestellte Zustand eingestellt und der Programmzylus gestartet.

Erster Zyklus mit aktiver Konfiguration

Im ersten Zyklus geschieht folgendes:

- Lokale Variablen und Funktionsblockvariablen werden initialisiert.
- RT_FIRST_SCAN wird für den ersten Zyklus des ersten ablaufenden Programms auf H-Pegel gesetzt.
- Die Logik wird bearbeitet.
- Die E/A wird aktualisiert und die Abfolge wiederholt.

Abschaltsequenz

Im letzten Zyklus geschieht folgendes:

- Das Programm empfängt keine Vorabmeldung über eine bevorstehende Abschaltung und kann keine speziellen Maßnahmen ergreifen.
- Die Schnittstellenmodule setzen den E/A-Sperrstatus entsprechend der jeweiligen Festlegung.

Normalbetrieb

Im Normalbetrieb geschieht folgendes:

- Alle mit einem Schritt verknüpften Aktionen werden in jedem Zyklus, in dem der Schritt aktiv ist, einmal ausgewertet (abhängig vom Aktions-Bestimmungszeichen).
- Aktionen werden ausgewertet, ehe der strukturierte Text innerhalb eines Schritts ausgewertet wird.
- Wird eine mit einem Schritt verknüpfte Transition TRUE, dann wird die E/A erneut aktiviert und danach alle entsprechenden Aktionen (nicht gepulst, inaktiv verzögert oder zeitbegrenzt (inaktiv)) ausgewertet, indem, mit Ausnahme von Haftmerkern, alle Ausgangsspulen abgeschaltet werden. Funktionsblöcke werden nicht ausgewertet. Die E/A wird aktualisiert und dann der nächste aktive Schritt im nachfolgenden Zyklus ausgewertet.
- Wird in einem Schritt strukturierter Text verwendet, dann wird die Transition erst ausgewertet, nachdem alle strukturierten Textanweisungen vollständig ausgeführt wurden.
- Alle strukturierten Textanweisungen innerhalb einer FOR-Schleife werden mit einer Iteration pro Zyklus ausgeführt, sofern keine Anweisung NOWAIT aufgetreten ist. In diesem Fall würde die Schleife alle Iterationen in einem einzigen Zyklus ausführen.
- Sämtliche strukturierte Textanweisungen in einem Schritt werden einmal während des ersten Zyklus ausgeführt, in dem der Schritt aktiv ist. Sie werden in nachfolgenden Zyklen nicht wiederholt.
- Alle Anweisungen innerhalb einer WHILE-Schleife werden solange ausgeführt, bis die WHILE-Bedingung FALSE wird.
- Die normale Bearbeitungsreihenfolge ist: E/A aktualisieren, Logik auswerten, dann wiederholen.

Dateinamen

In der nachstehenden Tabelle werden die einzelnen im Projektordner enthaltenen Dateitypen (Dateinamenerweiterungen) beschrieben:

_gprog.bbn	Binäre globale Konfigurationsdatei.
_gprog.st	Strukturierte Textversion der globalen Konfiguration.
CFG	Konfigurationsdatei
CSV	Variablendatei im CSV-Datenformat
DAT	Umgebungsparameter für Programmdateien
IL	Dateinnamenerweiterung für Anweisungslisten-Programme
OPI	Bedienerschnittstellen-Dateien.
R**	Dateinnamenerweiterung für Kontaktplanprogramme.
S**	Dateinnamenerweiterung für Ablaufplanprogramme
SBN	Von Runtimemaschine ausgeführte Binärdatei
SFR	Speichert die Wiederherstellungsdaten.
SFU	Speichert die Rückgängigdaten.
SFX	Speichert Daten zum Abbrechen von Online-Editiervorgängen.
SNF	Gemeinsam genutzte Namensdatei
SST	Strukturierte Textversion des Ablaufplanprogramms.
ST	Dateinnamenerweiterung für Programme in strukturiertem Text
SWC	Speichert die Namen der dem Überwachungsfenster hinzugefügten Symbole.

Hinweis

Dateien mit den Dateinamenerweiterungen RLL, OPI, SFC, CFG, ST, IL bilden die Quelle für Programme. Um eine Anwendung zu übertragen, brauchen Sie nur diese Dateien zu kopieren.

Kapitel 3 E/A konfigurieren

Nachdem ein Projekt erstellt wurde, wird im nächsten Schritt normalerweise für das Projekt eine Systemkonfiguration erstellt. Die Systemkonfiguration enthält eine Sammlung hardwarespezifischer Daten und Globalvariablen, die in allen Steuerungsprogrammen eines Projektes benutzt werden.

Ehe Sie die E/A Ihres Systems konfigurieren, sollten Sie möglichst die E/A-Schnittstellenplatine(n) und e E/A installieren. Sie können Ihr System aber auch konfigurieren, ohne zuvor die Hardware installiert zu haben. Klicken Sie in diesem Fall in der Platinen-Dialogbox bei den verwendeten E/A-Schnittstellenplatinen auf die Schaltfläche "Simulieren".

Informationen zur Installation und Konfiguration der Schnittstellenplatinen finden Sie in folgenden Anhängen:

E/A-System	Platine	Bestell- Nr.	Anhang
Genius	PCIM	IC660ELB921 (ein Port) IC660ELB922 (zwei Ports)	Anhang A
Series 90-30	PCIF1	IC693PIF301	Anhang B
Series 90-30	PCIF2	IC693PIF400	Anhang C

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Übersicht über System-E/A-Konfiguration
- Konfiguration der Genius E/A
- Konfiguration der Series 90-30 E/A
- Konfiguration anderer Feldblume
- Dynamischer Datenaustausch (DE)
- Konfiguration von Import/Export

3

Abschnitt 1: Übersicht

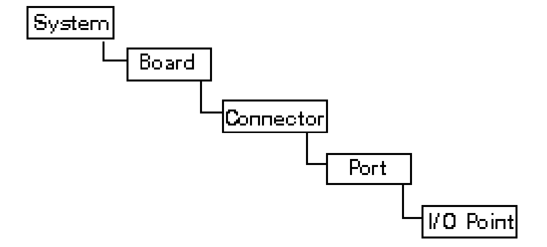
Die Dialogbox Systemkonfiguration enthält eine Sammlung mit hardwarespezifischen Informationen und Globalvariablen, die in einem Projekt in allen Steuerungsprogrammen verwendet werden.

Die Konfiguration teilt der Software mit, welche Schnittstellenmodule der Computer enthält und welche E/A an diesen Modulen angeschlossen ist. Mit dem Konfigurationseditor können Sie den E/A-Punkten Symbolnamen zuweisen. Sie können auch Kommentare zur Funktion der einzelnen Elemente eingeben.

Der Konfigurationseditor führt Sie schrittweise durch den Konfigurationsprozeß. Sie beginnen damit, dem System (dem Computer), an dem Sie arbeiten, einen Namen zuzuweisen und die Programmzykluszeit zu definieren. Hierauf folgen die PC-Steckplätze und die Steuerungssystemkarten (nicht die Computer-Grundmodule) in den einzelnen Steckplätzen, dann die an den Karten angeschlossenen E/A-Module und schließlich die E/A-Punkte.

Handelt es sich bei einer Karte um ein Kommunikationsmodul oder eine E/A-Zyklussteuerung, dann geht die Konfiguration noch tiefer und beschreibt dezentrale Chassis, Steckplätze, Platinen usw.

Die nachstehende Abbildung zeigt die Konfigurationsebenen für einfache On-Board-E/A:



Hilfe bei Hardwarekonflikten

Es gibt mehrere Einstellungen, die für den Datenaustausch mit den meisten Schnittstellenmodulen richtig konfiguriert werden müssen. Die gebräuchlichsten sind: Datenportadresse, Speichergrundadresse und Interrupt. Das Schnittstellenmodul kann nur dann richtig arbeiten, wenn die Brückeneinstellungen auf dem Modul mit den in der Dialogbox "Moduldefinition" gemachten Angaben übereinstimmen **UND** nicht mit der übrigen Hardware im PC in Konflikt geraten.

WICHTIG:

Am einfachsten können Sie die von den aktuellen Hardwaregeräten belegten Betriebsmittel herausfinden, indem Sie in den NT-Diagnosefunktionen in der Programmgruppe "Administrative Werkzeuge" nachschauen. Auf der Karteikarte "Betriebsmittel" werden die meisten der in Ihrem System benutzten Hardwareressourcen angezeigt. Denken Sie jedoch daran, daß einige der benutzten Geräte diese Benutzung möglicherweise nicht den NT-Diagnosefunktionen melden und daß derzeit nicht genutzte Geräte zwar nicht gemeldet sind, aber dennoch Konflikte verursachen können. Die NT-Diagnosefunktionen bilden aber einen guten Anfangspunkt.

Weitere Informationen zur Konfliktvermeidung finden Sie unter:

E/A-System	Platine	Bestell- Nr.	Anhang
Genius	PCIM	IC660ELB921 (ein Port) IC660ELB922 (zwei Ports)	Anhang A
Series 90-30	PCIF1	IC693PIF301	Anhang B
Series 90-30	PCIF2	IC693PIF400	Anhang C

Daten-E/A-Port

Der Daten-E/A-Port wird von den Gerätetreibern häufig für den Datenaustausch mit dem Modul verwendet.

Einige Portadressen sind bei den meisten PCs Standardeinstellungen. Die folgende Liste zeigt die gebräuchlichen Portverwendungen:
COM1: 3F8-3FF
COM2: 2F8-2FF
COM3: 3E8-3EF
COM4: 2E8-2EF
LPT1: 378-37A
Floppy: 3F0-3F7
Video: 3B0-3BB und 3C0-3DF

Speicheradresse

Mit der Speicheradresse wird ein gemeinsam benutzter Speicherbereich für das Schnittstellenmodul eingestellt. Normalerweise steht Speicher im Bereich C8000-DFFFF zur Verfügung. Dieser Speicherbereich wird normalerweise von Spezialmodulen wie dem Schnittstellenmodul benutzt. Zum Beispiel: Belegt das Module 4000 (hexa) Bytes gemeinsam genutzten Speicher ab D0000, dann ist der belegte Bereich D0000-D3FFF.

Interrupt – IRQ

Der Interrupt wird zur Kommunikation mit dem Modul verwendet. Zulässige Interruptwerte liegen zwischen 0 und 11.

Einige Interruptlevel sind Standardeinstellungen bei den meisten PCs und sollten nicht verwendet werden. Die nachstehende Liste zeigt die häufigsten Verwendungen:
COM1 und COM3: IRQ4
COM2 und COM4: IRQ3
Floppy: IRQ6
LPT1: IRQ7

IRQ1 und 2 werden immer von der Systemhardware verwendet.

Arbeiten mit Konfigurationsdateien

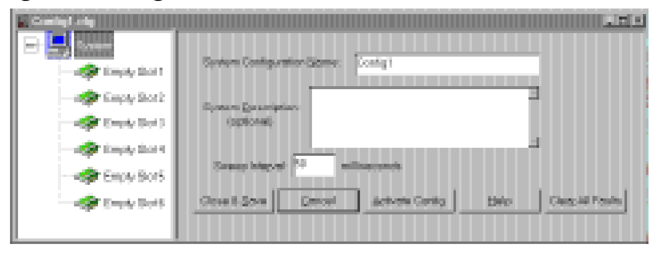
Elemente einer Konfiguration

Mit einer Konfiguration können Sie die E/A-Struktur definieren und E/A-PunktenE/A-Punkten und E/A-Ports Namensetikette zuweisen. Alle Konfigurationen haben einen Dateinamen mit der Erweiterung *.cfg.

Alle Konfigurationsoperationen werden vom Programmeditor aus durchgeführt.

Eine neue Konfiguration erstellen

1. Starten Sie den Programmeditor.
2. Klicken Sie auf *Datei* und wählen Sie *Neue Konfiguration*. Hierauf erscheint die Dialogbox "Konfiguration".



Eine bestehende Konfigurationsdatei bearbeiten

1. Starten Sie den Programmeditor.
2. Klicken Sie auf *Datei* und wählen Sie *Konfiguration öffnen*. Hierauf erscheint die Dialogbox "Konfiguration".
3. Bearbeiten Sie die Felder und klicken dann zuerst auf *Schließen* und danach auf *Speichern*.

Eine Konfigurationsdatei aktivieren

Klicken Sie auf *Projekt* und wählen dann *Konfiguration aktivieren*, um für das aktive Projekt eine neue Konfigurationsdatei zu aktivieren. Der Name der aktiven Konfigurationsdatei wird in der unteren Statusleiste des Programmeditors angezeigt.

Die aktive Datei wird von der Runtime-Umgebung zur Bestimmung der Struktur des E/A-Systems und globalen Symbolbenennung verwendet. Die aktive Konfigurationsdatei wird auch vom Programmeditor dafür verwendet, Ihnen auf

Systemebene verwendet Eingangs-, Ausgangs- und Speichersymbole während der Programmerstellung zur Verfügung zu stellen.

Bei der Aktivierung einer neuen Konfigurationsdatei kann es vorkommen, daß aktive Programme abgebrochen werden, wenn in der neuen Konfiguration keine Informationen aus der zuvor aktiven Konfiguration existieren. Die Aktivierung einer Konfigurationsdatei wird im aktiven Projekt eingetragen.

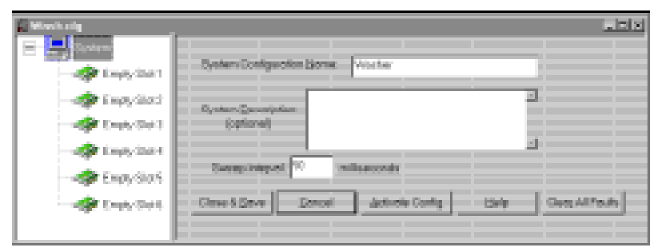
Jedesmal, wenn dieses Projekt geöffnet wird, wird auch die aktive Konfigurationsdatei für dieses Projekt aktiviert.

Navigieren im Konfigurations-Dienstprogramm

Systemkonfigurations-Dialogbox

Mit der Systemkonfigurations-Dialogbox können Sie der Konfiguration eine Systembezeichnung zuteilen. Diese Bezeichnung muß eine zulässige DOS-Bezeichnung aus maximal acht Zeichen sein. Es sind keine Leerzeichen zugelassen. Die Systembezeichnung in nachstehendem Beispiel ist "Washer”.

Standardmäßig werden Konfigurationsdateien im aktuellen Projektverzeichnis gespeichert.



E/A-Zykluszeit

Die E/A-Zykluszeit wird in der Systemkonfigurations-Dialogbox konfiguriert. Der E/A-Zyklus gibt an, in welchen Abständen die Steuerungsprogramme die E/A aktualisieren und die Programmlogik bearbeiten. Windows-Tasks mit geringerer Priorität werden vom E/A-Zyklus unterbrochen. In jedem E/A-Zyklus beginnt die Bearbeitung der aktiven Steuerungsprogramme, die dann bis zu ihrem Ende ablaufen.

Vor der Auswertung der Programmlogik liest die E/A-Zyklussteuerung die Eingangswerte. Nachdem die Steuerungslogik abgearbeitet wurde, werden die Ausgänge geschrieben. Die Ausführungszeit der Steuerungslogik ist variabel, da sich Anzahl und Typ der aktiven Anweisungen in den einzelnen E/A-Zyklen verändern können. Nachdem die Ausgänge aktualisiert wurden, wird die Steuerungstask angehalten und die NT-Tasks mit niedrigerer Priorität werden weiterbearbeitet.

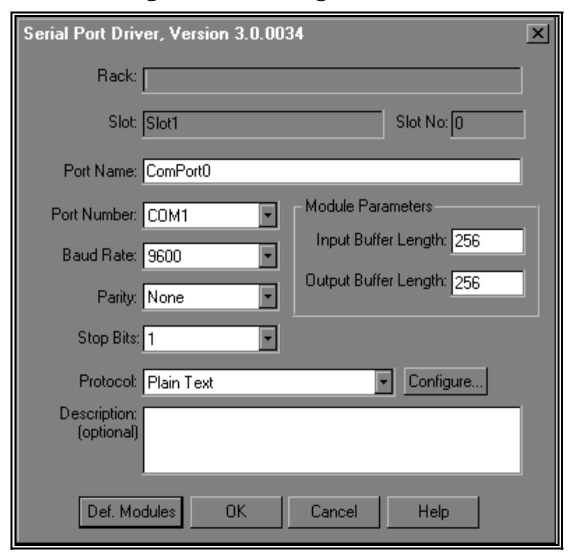
Bei der Definition einer Zykluszeit müssen Sie folgende Punkte beachten:

- Mindestdauer eines Eingabeimpulses - damit ein Eingangszustand von der Software erkannt werden kann, muß er für eine bestimmte Mindestdauer anstehen.
- Minimale Durchlaufzeit - die kürzeste Zeit, die das Steuerungssystem benötigt, um einen Wechsel des Ausgangssignals als Antwort auf einen Wechsel des Eingangssignals zu bewirken.
- Maximale Durchlaufzeit - die längste Zeit, die das Steuerungssystem benötigt, um einen Wechsel des Ausgangssignals als Antwort auf einen Wechsel des Eingangssignals zu bewirken.

Moduldefinitions-Dialogbox

Beiden einzelnen Modulen gibt es leichte Unterschiede in den Dialogboxen zur Konfiguration der E/A-Punkte. Genauere Angaben zur Konfiguration von E/A-Punkt finden Sie in der Hilfedatei des von Ihnen benutzten Treibers.

1. Klicken Sie in der Konfigurations-Dialogbox auf Moduldefinition.



2. Stellen Sie die Modulattribute entsprechend ein und klicken dann auf *OK*.

Abschnitt 2: GENIUS E/A konfigurieren

Das PC-Schnittstellenmodul (PCIM)

Mit dem PC-Schnittstellenmodul PCIM wird eine Verbindung zwischen einem PC und einem Genius E/A-System aufgebaut. Es gibt zwei verschiedene Arten von PCIMs: Einfachport und Doppelport. Jeder Port auf einem wird für den Datenverkehr auf einem bestimmten Genius E/A-Bus verwendet. Jeder Genius E/A-Bus kann bis zu 30 E/A-Blöcke unterhalten.

Mit dem PCIM-Moduldefinitionsdialog können Sie die Konfiguration der PCIM-Hardware und der E/A-Blöcke an den angeschlossenen Genius E/A-Bussen definieren.

PCIM konfigurieren

1. Klicken Sie auf die Schaltfläche "PCIM Konfigurations-Dienstprogramm aktivieren", wenn Sie das PCIM nicht mit dem PCIM Konfigurations-Dienstprogramm (PCU) konfiguriert haben. Hierdurch starten Sie ein eigenes Dienstprogramm, mit dem Sie eine Reihe von Konfigurationsparametern für das PCIM eingeben können. Die Online-Hilfe in diesem Dienstprogramm gibt Ihnen an, wie Sie das PCIM installieren und konfigurieren müssen.
2. Nachdem Sie das PCIM konfiguriert und das PCU verlassen haben, klicken Sie auf die Schaltfläche "PCIM Parameter aus Registrierdatenbank lesen". Hierdurch wird ein Fenster geöffnet, das eine Liste der konfigurierten PCIMs enthält.
3. Wählen Sie das entsprechende PCIM aus der Modellnummernliste aus.
4. Klicken Sie auf die Schaltfläche "PCIM Konfiguration testen", um die PCIM Konfiguration zu testen. Am PCIM müssen beide Lampen aufleuchten.

PCIM Moduldefinitions-Dialogbox

Name

Im Feld "Name" kann der Name eingetragen werden, unter dem das PCIM in der Konfiguration geführt wird. Mit diesem Namen wird das Modul in den Adressen der E/A-Punkte in der restlichen Software beschrieben.

Modellnummer

Es gibt zwei gültige Modelle: mit einem und mit zwei Ports. Während das Modul mit einem Port einen Genius E/A-Bus unterstützt, unterstützt das Modul mit zwei Ports zwei Busse. Bei einem Module mit einem Port können Sie über den PCIM Moduldefinitionsdialog nur die E/A und Hardwarekonfiguration für Port 1 definieren. Beidem Modul mit zwei Ports können Port 1 und Port 2 konfiguriert werden. Bei dem NULL-Modultyp kann keine E/A definiert werden.

Bit "Modul OK"

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert mit diesem Namen erzeugt. Dieser Boolesche Wert ist TRUE, wenn kein fehlerhaftes Modul erkannt wurde. Gibt es ein fehlerhaftes Modul, wird dieser Boolesche Wert FALSE.

Diskrepanz bei Konfiguration

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert mit diesem Namen erzeugt. Dieser Boolesche Wert ist TRUE, wenn die am Bus angeschlossenen Module nicht mit den in der Konfiguration eingestellten Werten übereinstimmen.

Beschreibung

Die Beschreibung ist eine frei formatierte textliche Beschreibung des Moduls. Dieses Feld muß in der Konfiguration nicht unbedingt ausgefüllt werden.

Ausgänge sperren wenn keine Programme laufen

Ist dieses Kästchen aktiviert, sind alle Ausgänge 0 wenn keine Programme laufen.

Port 1/Port 2

Kontrollkästchen "Simulieren"

Mit dem Kontrollkästchen "Simulieren" wird die Port-E/A gesperrt. Fehler- und Konfigurationsdaten werden ignoriert. Diese Funktion ist nützlich, wenn die E/A des Ports gerade nicht benutzt wird. Ist die E/A nicht simuliert, dann wird durch jeden von der Genius E/A-Blöcken erzeugte Fehler von der E/A-Zyklussteuerung ein Nothalt des aktiven Programms verursacht.

Netzwerknummer

Sie müssen eine beliebige Netzwerknummer zwischen 1 und 9 einstellen. Über die Netzwerknummer wird der gewünschte physikalische E/A-Bus mit dem in diesem Dialog definierten logischen Port verknüpft.

Achtung

Der PCIM E/A-Treiber kann nicht mehr als 9 E/A-Busse unterstützen. Jede in einer Konfiguration benutzte Netzwerksnummer muß eindeutig sein.

Interruptnummer

Der PCIM E/A-Treiber benutzt den in diesem Feld spezifizierten Interrupt, um Fehlermeldungen und Konfigurations-Änderungsmitteilungen für die E/A-Zyklussteuerung abzuwickeln. Der hier eingegebene Wert muß mit dem für das Modul angegebenen Wert übereinstimmen.

Gemeinsam genutzte RAM-Adresse

Das Feld "gemeinsam genutzte RAM-Adresse" gibt die Adresse des 16 k Puffers an, der für den Datenaustausch mit dem angegebenen Port auf dem PCIM verwendet wird. Hier werden nur zulässige Werte zur Auswahl angezeigt. Jede Adresse muß in der betreffenden Konfiguration eindeutig sein; zwei Ports können nicht gemeinsam den selben Puffer verwenden.

Der gesamte gemeinsam genutzte RAM-Adreßbereich darf von keinem anderen Gerät am PC verwendet werden. Es können dabei Konflikte auftreten die verhindern, daß das PCIM ordnungsgemäß funktioniert. Der hier angegebene Wert muß mit dem über die Einstellung der DIP-Schalter auf der Platine mit einem Port bzw. über das Konfigurationsprogramm für das Modul mit zwei Ports gewählten Wert übereinstimmen.

Hinweis

Die gemeinsam genutzte RAM-Adresse wird im PCIM Konfigurations-Dienstprogramm (PCU) eingestellt und hier angezeigt. Alle Änderungen der gemeinsam genutzten RAM-Adresse müssen im PCU durchgeführt werden.

E/A-Portadresse

Dieses Feld gibt die Anfangsadresse der vier zur Konfiguration und Steuerung des Ports verwendeten Bytes an. Die beiden letzten Bytes sind diejenigen, die über die

DIP-Schalter auf dem Modul mit zwei Ports eingestellt werden (z.B. würde ein Wert von 0x222 auf 0x220 abgebildet werden).

Die E/A-Portadresse muß mit dem über die DIP-Schalter für das Modul mit einem Port eingestellten Wert übereinstimmen und um 2 kleiner sein als der über die DIP-Schalter für das Modul mit zwei Ports eingestellten Wert.

PCIM Busadresse

Dieses Feld gibt die serielle Busadresse für diesen Port an.

Schaltfläche "Portkonfiguration testen"

Mit der Schaltfläche "Portkonfiguration testen" können Sie die in den Feldern Interruptnummer, gemeinsam genutzte RAM-Adresse und E/A-Portadresse gemachten Einträge überprüfen, indem Sie versuchen, das PCIM zu starten. Das Modul muß bei diesem Vorgang eingebaut sein.

Automatische Konfiguration von Netzwerkteilnehmern

Mit der Schaltfläche "Automatische Konfiguration von Netzwerkteilnehmern" wird der PCIM-Treiber veranlaßt, den Genius E/A-Bus nach allen Teilnehmern abzusuchen, die online sind. Die Teilnehmer werden automatisch in die Konfiguration eingetragen.

Schaltfläche "Globaldateneinstellung"

Mit dieser Schaltfläche können Sie für den Port Globaldaten einstellen. Hiermit wird die Dialogbox "Globaldateneinstellung" geöffnet.

Schaltfläche "PCIM E/A definieren"

Mit der Schaltfläche "PCIM E/A definieren" wird die Dialogbox "Genius Busadreßdefinitionen" geöffnet, in der Sie die Teilnehmer auf dem Genius E/A-Bus spezifizieren können.

Schaltfläche "OK"

Mit der Schaltfläche "OK" speichern Sie alle Änderungen ab und beenden die gesamte Editiersitzung für die Konfiguration. Sie kehren zurück zum Hauptmenü des Konfigurations-Dienstprogramms.

Schaltfläche "Abbrechen"

Mit der Schaltfläche "Abbrechen" verwerfen Sie alle Änderungen und beenden die gesamte Editiersitzung für die Konfiguration. Sie kehren zurück zum Hauptmenü des Konfigurations-Dienstprogramms.

Dialogbox "Globaldateneinstellung"

Am Genius Bus kann jeder Teilnehmen in jedem Zyklus bis zu 128 Datenbytes senden und von jedem anderen Teilnehmer am Bus bis zu 128 Datenbytes empfangen. Aktivieren Sie in dieser Dialogbox das Kästchen "Globaldaten freigeben" für jeden Teilnehmer von dem Sie Globaldaten empfangen wollen. Soll das PCIM Globaldaten zu anderen Teilnehmern senden, müssen Sie das Kästchen "Globaldaten freigeben" für den Teilnehmer aktivieren, zu dem die PCIMs Busadresse paßt.

Globaldateneinstellung

Drücken Sie diese Schaltfläche, um E/A-Markierungen einzurichten, die mit den Globaldaten für diese Busadresse verknüpft sind. Hierauf erscheint die Dialogbox "Globaldaten: Teilnehmer" für den selektierten Teilnehmer.

Globaldaten: Teilnehmer

Mit dieser Dialogbox können Sie den Globaldaten die über das PCIM gesendet und empfangen werden können, Namensetiketten zuweisen. Jedem Wort innerhalb des Globaldatenbereichs von 128 Bytes kann sein eigenes Etikett zugewiesen werden, und jedem Bit innerhalb der einzelnen Worte kann wiederum ein Etikett zugewiesen werden. Die aktuellen Namensetiketten der ersten 16 Worte werden am Bildschirm angezeigt. Verwenden Sie die Schaltflächen "nächste 16 Worte" und "vorherige 16 Worte", um die Namen der übrigen Worte zu bearbeiten. Neben jedem Namensetikett finden Sie ein Kontrollkästchen. Um einzelnen Bits innerhalb eines Wortes Namensetikette zuzuweisen, müssen Sie das Kästchen "Bits benennen" neben dem Wort aktivieren, dessen Bits mit einen Namen versehen werden sollen, und dann die Schaltfläche "Bitnamen editieren" drücken. Hierauf erscheint die Dialogbox "Globaldaten: Teilnehmer, Wort" für den selektierten Teilnehmer.

Globaldaten: Teilnehmer, Wort

Mit dieser Dialogbox können Sie einzelnen Bits innerhalb des selektierten Wortes Namensetiketten zuweisen.

Genius Busadreßdefinitionen

Mit der Dialogbox "Genius Busadreßdefinition" werden die auf einem Genius E/A-Bus definierten E/A-Blöcke eingegeben und bearbeitet. Jeder E/A-Block wird in einer Adresse am Genius E/A-Bus definiert. Der zulässige Adreßbereich geht von 0 bis 31. Die definierten Blöcke sind im Bildlaufbereich aufgeführt.

Verwendung der Dialogbox "Genius Busadreßdefinitionen"

Teilnehmer-Konfigurationstabelle für diesen Port freigeben

Gibt die Teilnehmer-Konfigurationstabelle frei.

Datagramme für diesen Port freigeben

Gibt die Datagramme für diesen Port frei. Weitere Informationen finden Sie unter "Datagramme mit dem PCIM-Treiber senden”.

Bus

Zeigt die Buskennung an.

Blocktyp

In diesem Feld wird der definierte E/A-Blocktyp angezeigt. (Der Standard-Blocktyp für Bus ID 31 ist PCIM; er kann nicht verändert werden).

Blockname

Dieses Feld zeigt den Namen eines E/A-Blocks an.

E/A-Gerät auswählen

Diese Schaltfläche öffnet die Dialogbox "Blockdaten", in der Sie den Typ des E/A-Blocks an der momentan gewählten Adresse verändern können. Um einen E/A-Block vom Bus zu entfernen, selektieren Sie den Block, drücken diese Schaltfläche und wählen NULL als neuen Blocktyp.

Schaltfläche "Ports definieren"

Die Schaltfläche "Block definieren" bewirkt die Anzeige der mit dem aktuell selektierten E/A-Block verknüpften Dialogbox. In dieser Dialogbox können Sie die

E/A-Punkte des angegebenen E/A-Blocks bearbeiten. Auf einen Block doppelklicken hat die gleiche Wirkung wie wenn Sie einen Block selektieren und diese Schaltfläche drücken.

Schaltfläche "Zurück zu Modul"

Mit dieser Schaltfläche kehren Sie zurück zum PCIM Moduldefinitionsdialog.

Schaltfläche "OK"

Mit der Schaltfläche "OK" speichern Sie alle Änderungen ab und beenden die gesamte Editiersitzung für die Konfiguration. Sie kehren zurück zum Hauptmenü des Konfigurations-Dienstprogramms.

Schaltfläche "Abbrechen"

Zeigt die Dialogbox "Abbruch bestätigen" an. Wenn Sie mit "Ja" antworten, wird die Editiersitzung beendet und Ihre Konfigurationsänderungen werden nicht gespeichert.

Diskrete Punktdaten

Mit dem Dialog "Diskrete Punktdaten" können Sie die mit einem Genius E/A-Block verbundenen E/A-Punkte definieren.

Statusbit-Name

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert mit diesem Namen erzeugt. Ist der Block fehlerfrei, dann ist dieser Wert TRUE. Enthält der Block einen Fehler, dann ist dieser Wert FALSE.

Eingang gesamter Block/Ausgang gesamter Block

Eines dieser Felder oder beide werden angezeigt. Wird in dieses Feld ein Name eingetragen, dann wird eine Variable erzeugt, deren Größe und Typ den gesamten Block abdeckt. Handelt es sich bei dem Block zum Beispiel um eine diskreten 16-Bit Eingangsblock, dann wird ein Eingangs-WORT erzeugt. Bei Modulen mit konfigurierbaren Punktrichtungen dürfen nur die Bits von dem Symbol "Eingang gesamter Block" gelesen werden, die Eingangspunkten entsprechen. Ebenso dürfen nur Bits, die Ausgangspunkten entsprechen, nach "Ausgang gesamter Block" geschrieben werden.

Bit

Zeigt die Bitnummer im E/A-Block an. Zulässige Werte liegen zwischen 1 und 16. Sie können diese Werte nicht verändern, sie dienen nur zur Information.

Name

Das Feld "Name" enthält den anwenderdefinierten Namen des E/A-Punktes. Dieser Name muß den Namensrichtlinien der Norm IEC-1131 entsprechen. Der Name muß mit einem Buchstaben beginnen und darf alphanumerische Zeichen und Unterstriche enthalten. Der Name eines E/A-Punktes darf keine Leerzeichen enthalten. Beachten Sie, daß bei den Namen zwischen Groß- und Kleinbuchstaben unterschieden wird.

Schaltfläche "Richtung"

Gestattet der Blocktyp eine Konfiguration der Punktrichtung, dann bewirkt ein Klicken auf die Schaltfläche "Richtung", daß die Schaltfläche zwischen "Eingang" und"Ausgang" umgeschaltet wird.

Schaltfläche "Zurück zu Bus"

Wenn Sie auf die Schaltfläche "Zurück zu Bus" klicken, werden die Änderungen gespeichert. Sie kehren zurück zum Dialog "Genius Busadreßdefinition".

Analoge Punktdaten

Mit dem Dialog "Analoge Punktdaten" können Sie die mit einem Genius E/A-Block verbundenen E/A-Punkte definieren.

Statusbit-Name

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert mit diesem Namen erzeugt. Ist der Block fehlerfrei, dann ist dieser Wert TRUE. Enthält der Block einen Fehler, dann ist dieser Wert FALSE.

Name

Das Feld "Name" enthält den anwenderdefinierten Namen des E/A-Punktes. Dieser Name muß den Namensrichtlinien der Norm IEC-1131 entsprechen. Der Name muß mit einem Buchstaben beginnen und darf alphanumerische Zeichen und Unterstriche enthalten. Der Name eines E/A-Punktes darf keine Leerzeichen

enthalten. Beachten Sie, daß bei den Namen zwischen Groß- und Kleinbuchstaben unterschieden wird.

Schaltfläche "Richtung"

Diese Schaltfläche dient nur zur Information. Die Richtung von Analogpunkten wird durch den Typ des E/A-Blocks festgelegt.

Schaltfläche "Zurück zu Bus"

Wenn Sie auf die Schaltfläche "Zurück zu Bus" klicken, werden die Änderungen gespeichert. Sie kehren zurück zum Dialog "Genius Busadreßdefinition".

Datagramme mit dem PCIM-Treiber senden

Im Dialog "Busadreßdefinition" gibt es für diesen Port ein Kontrollkästchen "Datagramme freigeben".

Ist dieses Kästchen aktiviert, werden 11 Symbole in der Datei .cfg angelegt. Sie haben folgendes Präfix:

<Ihr Modulname>_Network<Ihre Netzwerknummer>_

Diese Symbole entsprechen dem Befehlsblock auf der µGeni-Platine des PCIM (siehe Beschreibung in Kapitel 10 des *µGeni-Platine Anwenderhandbuchs*, GFK-0845). Die einzigen Befehle, die so ausgegeben werden sollen, sind "Datagramm senden" und "Datagramm senden mit Antwort".

Ehe DGramStatus auf 1 gesetzt wird, müssen alle Felder ausgefüllt werden. Danach verhält sich die Variable DGramStatus so, wie dies im *µGeni-Platine Anwenderhandbuch* beschrieben wird. Informationen zu Datagramm-Datenformaten finden Sie in *Genius E/A-System und Datenübertragung, Anwenderhandbuch*, GEK-908486-1.

Symbol	Beschreibung
DGramStatus	Status
DGramCmdType	Befehlstyp (02="Datagramm senden", 03=Senden mit Antwort)
DGramDestAddr	Ziel-Busadresse
DGramFunc	Funktionscode
DGramSubFunc	Unterfunktionscode
DgramHdrByte5	Siehe nachstehende Erläuterung.
DgramHdrByte6	Siehe nachstehende Erläuterung.
DgramHdrByte7	Siehe nachstehende Erläuterung.
DgramHdrByte8	Siehe nachstehende Erläuterung.
DGramData	BYTE-Feld (ausgehende Daten)
DGramReplyData	BYTE-Feld (Antwortdaten)

Bei der Ausgabe eines Befehls "Datagramm senden" ist HdrByte5 die Priorität und HdrByte6 die Länge.

Bei der Ausgabe eines Befehls "Datagramm senden mit Antwort" ist HdrByte5 der Unterfunktionscode des Antwort-Datagramms, HdrByte6 die Priorität, HdrByte7 die Länge des Sendedatenpuffers und HdrByte8 die Länge des Antwortdatenpuffers (von der µGeni-Platine eingestellt).

Abschnitt 3: Konfiguration der Series 90-30 E/A

Die PC Control Software unterstützt folgende E/A-Module der Series 90-30.

Bezeichnung in PC Control	Bestellnummer	Einzelheiten zur Installation
PCIF1	IC693PIF301	Anhang B
PCIF2	IC693PIF400	Anhang C

Die PC-Schnittstelle (PCIF)

Mit dem PC-Schnittstellenmodul PCIF wird eine Verbindung zwischen einem PC und einem Series 90-30 E/A-System aufgebaut. Mit dem PCIF-Moduldefinitionsdialog wird die Konfiguration der PCIF-Hardware und der Series 90-30 E/A definiert.

Verwendung des PCIF-Moduldialogs

Modulname

Geben Sie für diese Karte einen Namen ein. Sie können die Standardbezeichnung belassen oder einen neuen Namen eingeben. Dieser Name wird im Symbolmanager angezeigt und hilft bei der Identifikation der Lage der E/A-Punkte.

Portadresse und Modulnummer

Mit diesen Listenfeldern geben Sie die E/A-Grundadresse und Modulnummer Ihrer PCIF-Karte in den PC ein. Die Grundadresse muß mit den Einstellungen der DIP-Schalter auf der Karte übereinstimmen. Jede PCIF-Karte muß eine eindeutige Modulnummer besitzen, die mit 0 beginnt. Informationen zur Installation der PCIF-Karte und zur Auswahl einer geeigneten E/A-Grundadresse finden Sie in Anhang B (PCIF1) oder Anhang C (PCIF2), oder wenn Sie auf die Hilfetaste neben dem Feld "Grundadresse" klicken.

Kontrollkästchen "Simulieren"

Aktivieren Sie dieses Kästchen, um Anwenderprogramme laufen zu lassen, ohne auf das E/A-System zuzugreifen. Dies ist hilfreich bei Test- und Entwicklungsarbeiten, wenn PCIF-Karte oder E/A-Chassis nicht verfügbar sind.

Ausgänge sperren wenn keine Programme laufen

Aktivieren Sie dieses Kästchen, um alle Ausgänge zu löschen wenn keine Programme laufen. Dies kann eine wichtige Sicherheitsfunktion bedeuten. Deaktivieren Sie dieses Kästchen, wenn Sie die Ausgänge am Programmende aktiv belassen wollen oder wenn Sie die Ausgänge auf EIN fixieren wollen, ohne daß dabei ein Programm läuft.

Programm bei Fehler abbrechen

Ist dieses Kästchen aktiviert, dann werden zwar alle Fehler am Bildschirm gemeldet und die Statusbits weiterhin richtig gesetzt, die laufenden Programme werden aber weiterhin bearbeitet. Ist dieses Kästchen nicht aktiviert, dann werden die Programme abgebrochen, wenn ein Fehler auftritt.

Bitname für Diskrepanz bei Konfiguration

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert erzeugt. Dieser Wert wird auf TRUE gesetzt, wenn die Konfiguration nicht mit der aktuellen Hardware übereinstimmt.

Statusbit-Name

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert erzeugt. Dieser Wert ist TRUE, wenn alle an dieser Platine angeschlossenen Module in Ordnung sind. Er wird FALSE, wenn ein Modul fehlerhaft ist.

Beschreibung: (wahlweise)

Dieser Platz steht für eine Wahlfreie Beschreibung der E/A-Karte.

Chassis

Diese Schaltflächen öffnen Dialoge zur Konfiguration der E/A-Module in den einzelnen Chassis. PCIF1 unterstützt vier Chassis; PCIF2 unterstützt sieben Chassis. Die Schaltflächen sind grau, bis jedes Chassis definiert ist.

90-30 Chassis definieren

Mit dieser Schaltfläche öffnen Sie eine Dialogbox zur Definition der an der PCIF-Karte angeschlossenen E/A-Chassis.

Automatische Konfiguration

Diese Schaltfläche aktiviert die Funktion "Automatische Konfiguration". Sie können diese Funktion einsetzen, wenn E/A-Chassis und Module eingebaut und eingeschaltet sind und das Runtime-Subsystem nicht aktiv ist. Wenn Sie auf die Schaltfläche Automatische Konfiguration klicken, dann aktiviert die Software die PCIF-Karte und liest die Informationen über angeschlossene Chassis und Module.

Modul einstellen und testen

Diese Schaltfläche kommuniziert mit der PCIF-Karte um zu überprüfen, ob die Installation in Ordnung ist. (Das Runtime-Subsystem darf nicht aktiv sein). Es sollte eine Dialogbox "Modultest" erscheinen, die die Meldung "PCIF2 Board found, shared RAM access successfully found” [PCIF2-Modul gefunden, auf gemeinsam genutzten RAM erfolgreich zugegriffen] enthält. Erscheint diese Meldung nicht, sollten Sie versuchen, einen anderen Block gemeinsam genutztem RAM zu konfigurieren.

Der Chassisdefinitionsdialog

Mit der Dialogbox "Chassisdefinition" können Sie das an der PCIF-Karte angeschlossene Chassis von Hand konfigurieren. Geben Sie nur eine Chassibezeichnung ein und wählen den Chassistyp aus.

Mit den Modulschalltflächen werden die Moduldefinitionsdialoge für die zugehörigen Chassis aufgerufen.

Der Moduldialog

Modulname

Geben Sie für das Modul einen beschreibenden Namen ein. Dieser Name wird im Symbolmanager angezeigt und hilft bei der Suche nach einem bestimmten E/A-Symbol.

Modulposition

Dies ist die Nummer des physikalischen Steckplatzes im Chassis. Verwenden Sie die Schaltflächen "Nächstes Modul" und "Vorheriges Modul", um zwischen Modulen im gleichen Chassis umzuschalten.

Modulkennung

Geben Sie die Modulkennung des Moduls in diesem Steckplatz ein. Drücken Sie die Schaltfläche "Nachschlagen", um die Modul-Modellnummer im Beschreibungsfeld anzuzeigen.

Schaltfläche "Ports"

Diese Schaltfläche öffnet einen Dialog zur Definition der einzelnen Ports auf diesem Modul.

Definition der digitalen Portanschlüsse

Geben Sie für jedes benutzte E/A-Bit einen Symbolnamen ein. Verwenden Sie die Schaltflächen "Nächster Anschluß" bzw. "Vorheriger Anschluß", um zwischen den E/A-Gruppen auf diesem Modul umzuschalten.

Das Kästchen oben im Dialog zeigt die Richtung der E/A an (Eingang oder Ausgang).

Modul-Störungsbit

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert erzeugt. Dieser Wert wird TRUE, wenn das Modul in Ordnung ist. Er wird FALSE, wenn im Modul ein Fehler erkannt wird.

Definition der analogen Portanschlüsse

Geben Sie für jeden aufgelisteten Analogport einen beschreibenden Namen ein. Jeder Name muß für die gesamte globale Symboldatenbank eindeutig sein.

Modul-Störungsbit

Wenn Sie in dieses Feld einen Namen eingeben, wird eine Boolescher Wert erzeugt. Dieser Wert wird TRUE, wenn das Modul in Ordnung ist. Er wird FALSE, wenn im Modul ein Fehler erkannt wird.

Abschnitt 4: Konfiguration anderer Feldbusse

PC Control unterstützt den Anschluß zu den gebräuchlichsten Bustypen. Dieser Abschnitt beschreibt die Konfiguration der Parameter für den Anschluß an DeviceNet und Profibus.

Konfiguration von DeviceNet

Mit dem DeviceNet-Treiber können Sie mit bis zu 63 Knoten eines DeviceNet-Netzwerks Daten austauschen. Der Treiber besitzt eine offene Konfigurationsschnittstelle, die alle Geräte mit ODVA-Zulassung unterstützt. Eine Vielzahl verschiedener E/A-Geräte steht zur Befriedigung Ihrer Steuerungsanforderungen zur Verfügung.

Der DeviceNet-Treiber benutzt das Scannermodul 5136-DN von S-S Technologies. Diese Karte führt die Abfrage der E/A-Geräte am Netzwerk durch.

Leistungsmerkmale

DeviceNet unterstützt bis zu 64 Netzknoten. Wird für das 5136-DN Modul eine MAC ID reserviert, dann bleiben noch 63 Knoten für den Einsatz in Anwendungen übrig. Der Treiber unterstützt sowohl zyklisch als auch bitweise abgefragte E/A-Verbindungen. Explizite Meldungsübermittlung ist nicht unterstützt. Es werden Netzwerksgeschwindigkeiten von 125, 250 und 500 kBd unterstützt.

Der Treiber konfiguriert das 5136-DN Scannermodul so, daß es als DeviceNet Master arbeitet. Ein DeviceNet Master "besitzt" die Slaves, deren MAC IDs in seiner Abfrageliste stehen. Mit Ausnahme einer Überprüfung auf doppelt vorkommende MAC IDs, kann ein Slave keine Kommunikation beginnen, wenn er nicht von seinem Master dazu aufgefordert wurde. Das 5136-DN wird als E/A-Scanner eingestellt, der alle freigegebenen Teilnehmer in der Treiberkonfiguration bedient.

Installation und Konfiguration von Teilnehmern am DeviceNet Netzwerk

Einzelheiten zur Installation eines bestimmten Gerätes am DeviceNet Netzwerk finden Sie in der Bedienungsanleitung zu diesem Gerät. Zur Vermeidung von Adressenkonflikten müssen Sie darauf achten, daß die MAC ID des Teilnehmers richtig eingestellt ist. Zahlreiche einfache Geräte können Sie über DIP-Schalter konfigurieren. Komplexere Geräte werden online über das Netzwerk konfiguriert. Zur ordnungsgemäßen Konfiguration benötigen solche Geräte ein DeviceNet Management-Dienstprogramm. GE Fanuc empfiehlt die Verwendung eines ODVA-zugelassenen Softwarepakets zur Konfiguration Ihres Geräts. Lassen Sie sich von Ihrem DeviceNet Distributor eine Liste der Lieferanten von Netzwerk-Managementsoftware geben.

Zur Konfiguration Ihres DeviceNet Netzwerks gibt es zahlreiche unterschiedliche Boxen. In diesem Abschnitt werden die Dialogboxen und deren Funktion zusammengefaßt. Ausführliche Angaben zu den einzelnen Dialogboxen finden Sie in der Online-Hilfe..

Dialogbox:	Beschreibung
Dialogbox "Modul"	In der Dialogbox "Modul" können Sie das 5136-DN Scannermodul für Ihr DeviceNet Netzwerk konfigurieren.
Dialogbox "Fehlererkennung"	Mit der Dialogbox "Fehlererkennung" können Sie Netzwerk und Gerätefehler auswählen.
Dialogbox "DeviceNet Netzknoten bearbeiten"	Mit der Dialogbox "DeviceNet Netzknoten bearbeiten" können Sie die einzelnen Teilnehmer im Netzwerk auswählen und bearbeiten.
Dialogbox "Teilnehmer bearbeiten"	Mit der Dialogbox "Teilnehmer bearbeiten" können Sie einen DeviceNet Teilnehmer an der angegebenen MAC ID bearbeiten.
Dialogbox "E/A-Datenstrom-Editor"	Mit dem E/A-Datenstrom-Editordialog können Sie das exakte Format der Ein- und Ausgangs-Datenströme des ausgewählten Teilnehmers definieren. Sie können auch an die einzelnen Stromkomponenten symbolische Informationen anhängen.
Dialogbox "Datenstromkomponente bearbeiten"	Mit dem Dialog "Datenstromkomponente bearbeiten" können Sie der ausgewählten Datenstromkomponente einen symbolischen Namen zuweisen und ihren Typ definieren.
Globalvariablen	Werden in der DeviceNet Konfiguration bestimmte Symbolnamen definiert, liefert der Treiber automatisch einen Satz von Globalvariablen, die Status- und Fehlerinformationen melden. Wird ein Modulname angegeben, erstellt der Treiber einen Satz Globalvariablen, die Status- und Fehlerinformationen auf dem CAN-Bus liefern. Wird ein Teilnehmer-Symbolname definiert, liefert der Treiber eine Variable, die den Status dieses Teilnehmers meldet.
CAN-Bus Globalvariable	Wird im Moduldialog ein Modulname angegeben, erstellt der Treiber automatisch einen Satz Globalvariablen, die Status- und Fehlerinformationen auf dem CAN-Bus liefern. Zur Erzeugung der einzelnen Globalvariablen wird symbolische Information am Ende des Modulnamens angehängt. Wird der Modulname im Dialog gelöscht, wird die Erzeugung dieser Variablen gesperrt.
Teilnehmerstatus Globalvariable	Wurde im Feld "Symbolname" des Dialogs "Teilnehmer bearbeiten" ein Teilnehmername definiert, dann erzeugt der Treiber automatisch eine Teilnehmer-Statusvariable. Zur Erzeugung der einzelnen Globalvariablen wird symbolische Information am Ende des Teilnehmernamens angehängt. Wird der Symbolname im Dialog ""Teilnehmer bearbeiten" gelöscht, dann wird die Erzeugung dieser Variablen gesperrt.

Konfiguration der Profibus E/A

PC Control verwendet ein Hilscher CIF30/CIF104, CIF30_FMS oder CIF30_PB Modul zum Datenaustausch mit einem Profibus Feldbus-Netzwerk.

Die Konfiguration des ProfiBus Treibers erfolgt in zwei Schritten. Zuerst ist eine Konfiguration in der SyconDP Konfigurationssoftware, einem Produkt von Synergetic Microsystem, erforderlich. Anschließend müssen alle Konfigurationsdaten in ähnlicher Weise in PC Control eingetragen werden. Die Konfiguration eines ProfiBus Netzwerks in der SyconDP Software wird nachstehend ausführlich beschrieben.

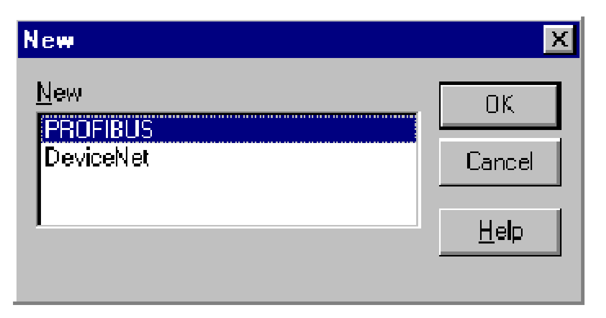
Ehe Sie mit der Arbeit beginnen müssen Sie sicherstellen, daß die .GSD-Dateien, die benutzt werden sollen, im entsprechenden Verzeichnis installiert wurden:

```
Program Files\Hilscher GmbH\SyCon\Fieldbus\Profibus\GSD
```

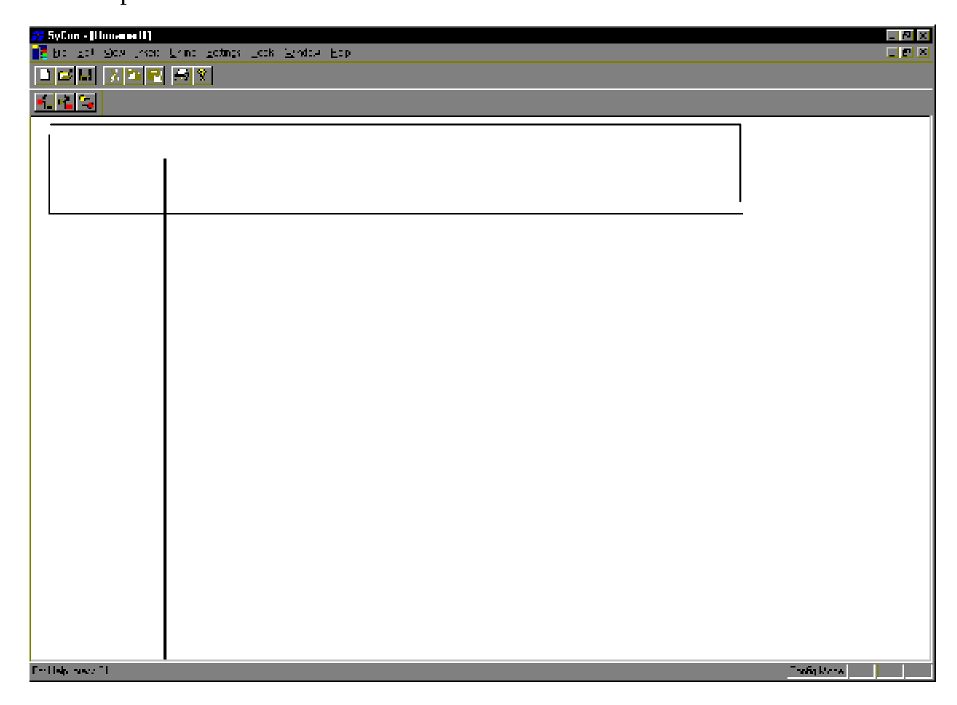
Die benötigten Dateien sind:

gef_0534.gsd	Hil_7506.gsd
Hil_7501.gsd	Hil_7507.gsd
Hil_7502.gsd	Hms_1002.gsd
Hil_7503.gsd	UN_COMBI.gsd
Hil_7504.gsd	UN_DP.gsd
Hil_7505.gsd	VMXM0534.gsd

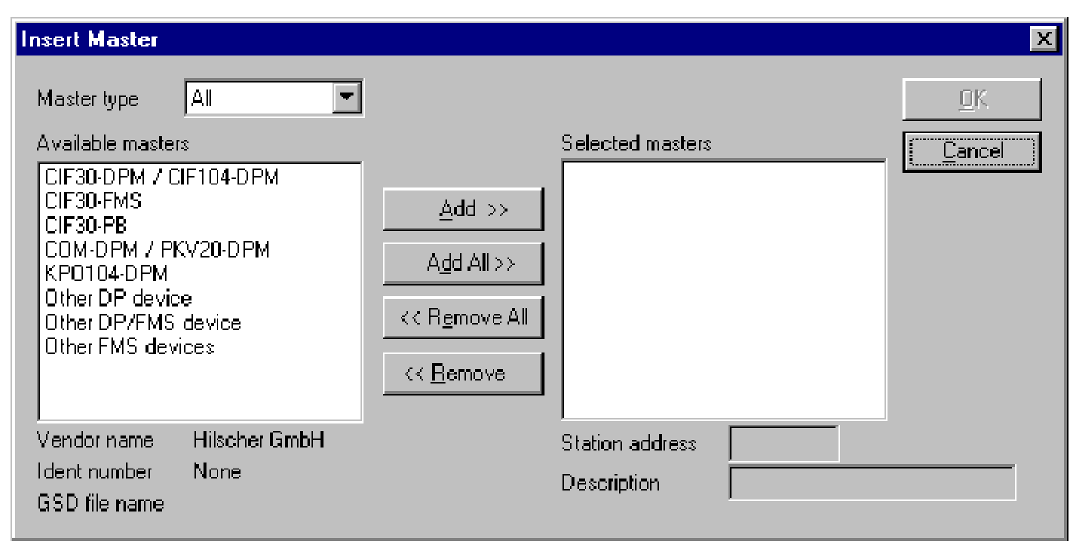
1. Starten Sie die SyCon SYstem CONfigurator Software und erstellen Sie eine neue Datei. Hierauf erscheint die Dialogbox "Neu".



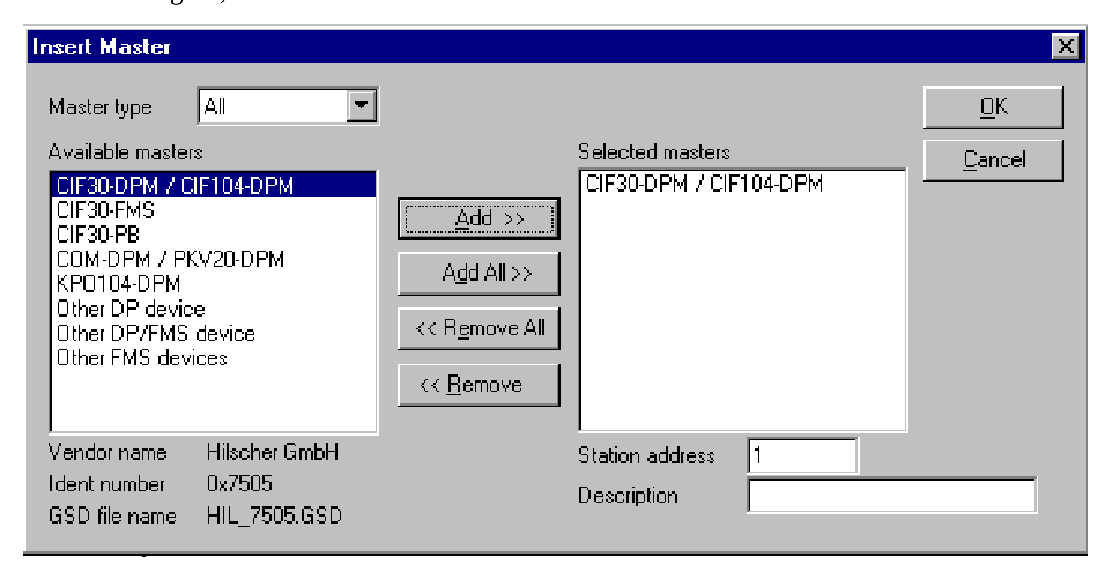
2. Selektieren Sie PROFIBUS und drücken dann OK. Hierauf erscheint das Haupt-Netzwerkmenü.



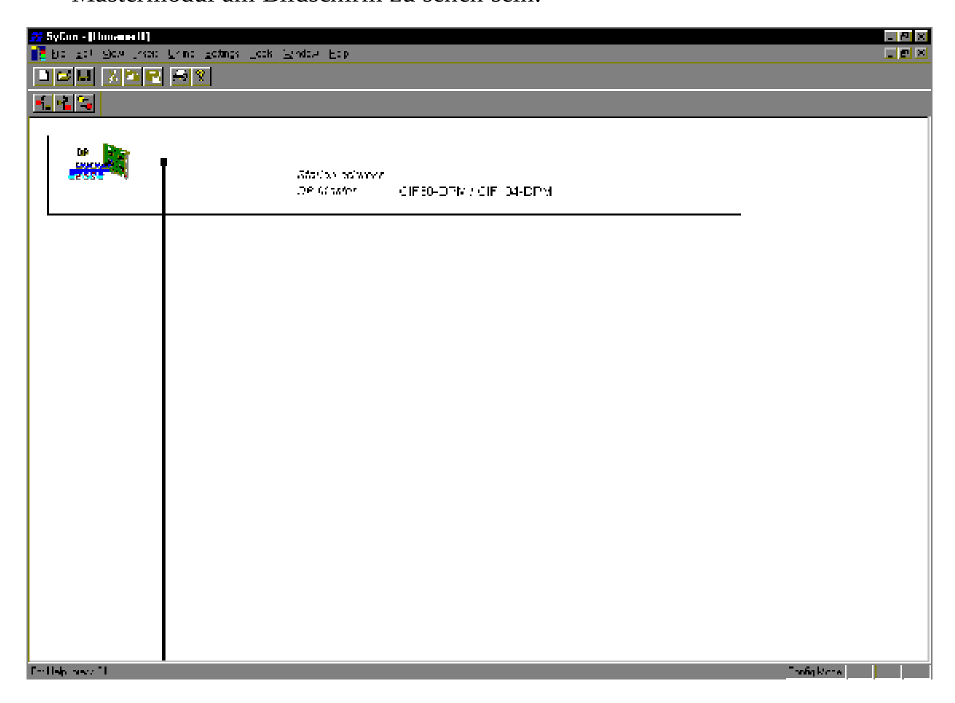
3. Wählen Sie die Schaltfläche ganz links auf der Symbolleiste, um den Master einzufügen, oder wählen Sie "Einfügen" und dann "Master" aus dem Pull-Down-Menü. Setzen Sie den Cursor über das Rechteck und klicken Sie die linke Maustaste. Hierauf erscheint die Dialogbox "Master einfügen".



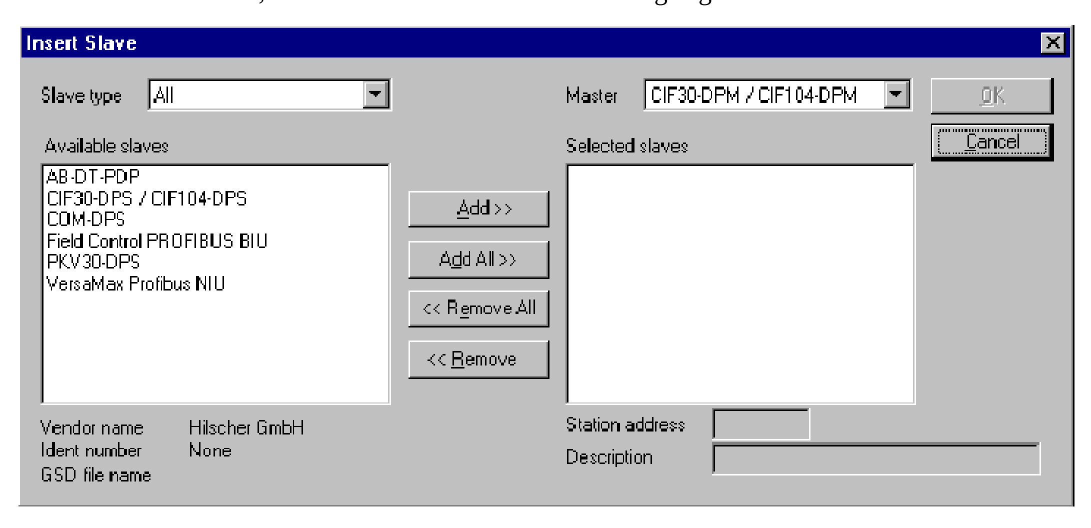
4. Wählen Sie Ihr Mastermodul und klicken dann erst auf die Schaltfläche "Hinzufügen", dann auf OK.



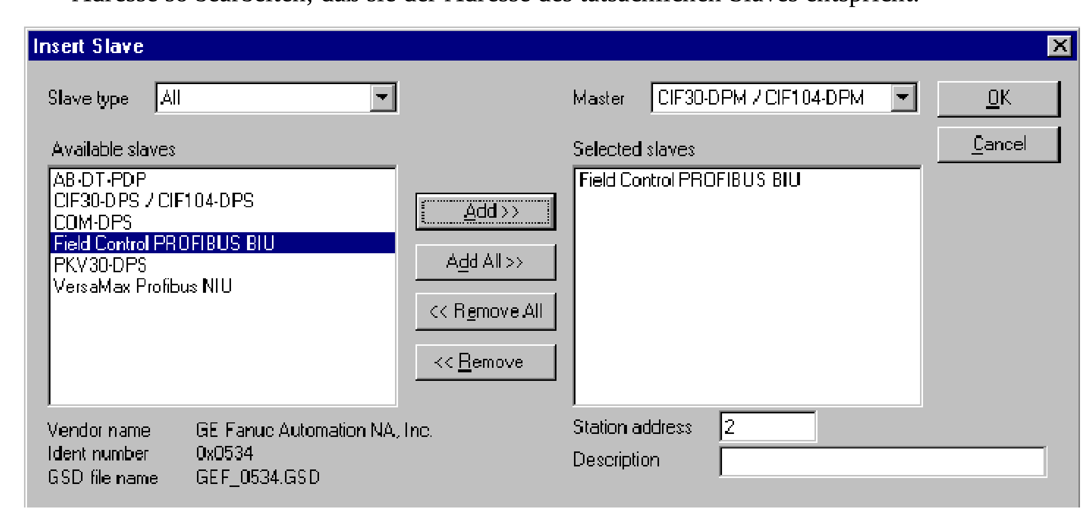
Jetzt sollte das nachstehende Netzwerkmenü mit dem ausgewählten Mastermodul am Bildschirm zu sehen sein.



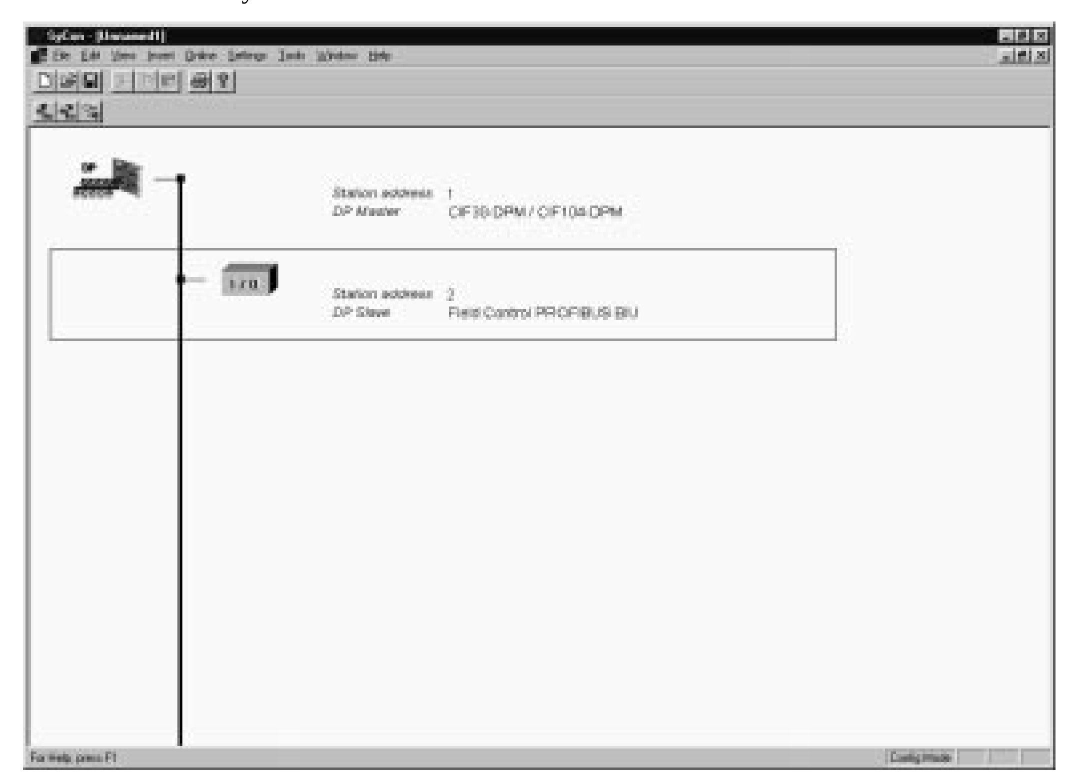
5. Wählen Sie die Schaltfläche in der Symbolleiste, um den Slave einzufügen, oder wählen Sie "Einfügen" und dann "Slave" aus dem Pull-Down-Menü. Setzen Sie den Cursor unter das Rechteck und klicken Sie die linke Maustaste. Jetzt sollte das folgende Menü angezeigt werden mit den verfügbaren Slaves für alle GSD-Dateien, die Sie im Verzeichnis \Profibus abgelegt haben.



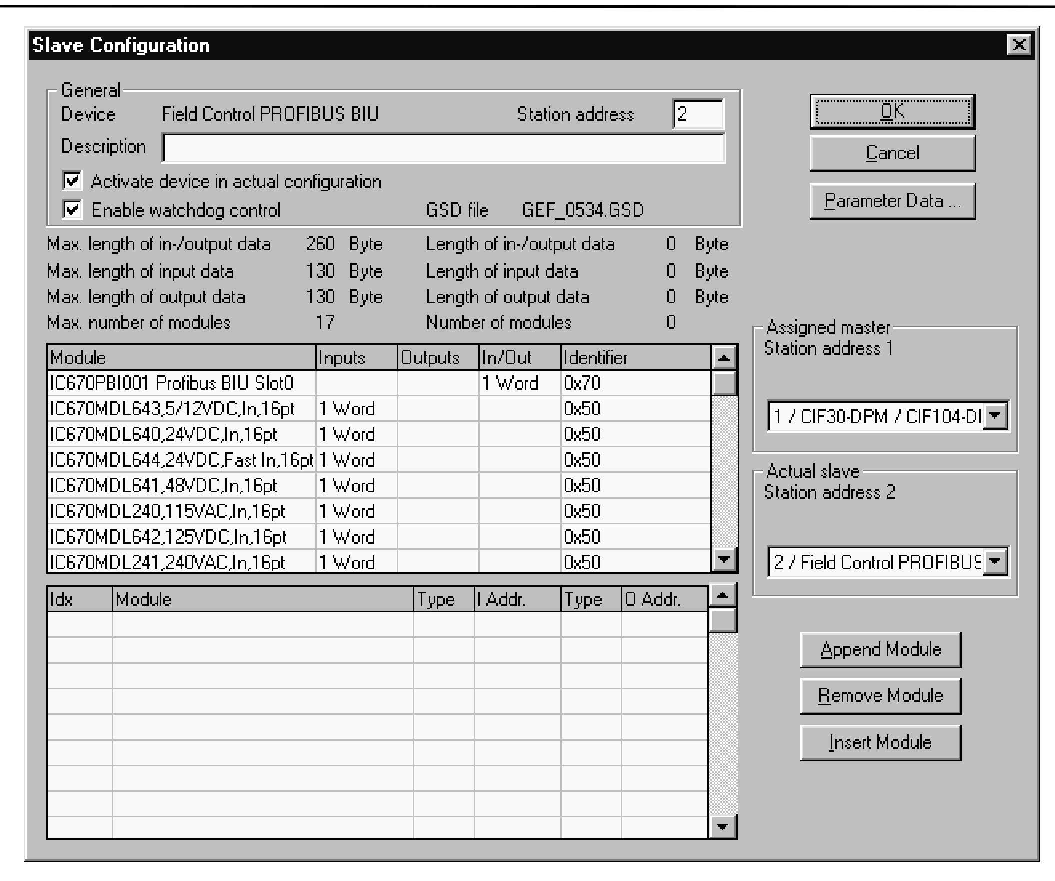
6. Wählen Sie den entsprechenden Slave-Teilnehmer und drücken die Schaltfläche "Hinzufügen". Die Adresse des Teilnehmers am Profibus wird automatisch auf die nächste verfügbare Adresse erhöht. Sie können jetzt die Adresse so bearbeiten, daß sie der Adresse des tatsächlichen Slaves entspricht.



Nachdem Sie auf OK geklickt haben, sollten Sie jetzt das nachstehende Netzwerkmenü sehen. Wiederholen Sie diesen Schritt für jeden Slave in Ihrem System.



7. Sie können jetzt die Konfiguration Ihres Slaves bearbeiten. Doppelklicken Sie auf den Teilnehmer, den Sie bearbeiten wollen. Hierauf erscheint eine Dialogbox ähnlich der nachstehenden. Die Daten in Ihrer Dialogbox müssen der GSD-Datei des ausgewählten Slaves entsprechen.

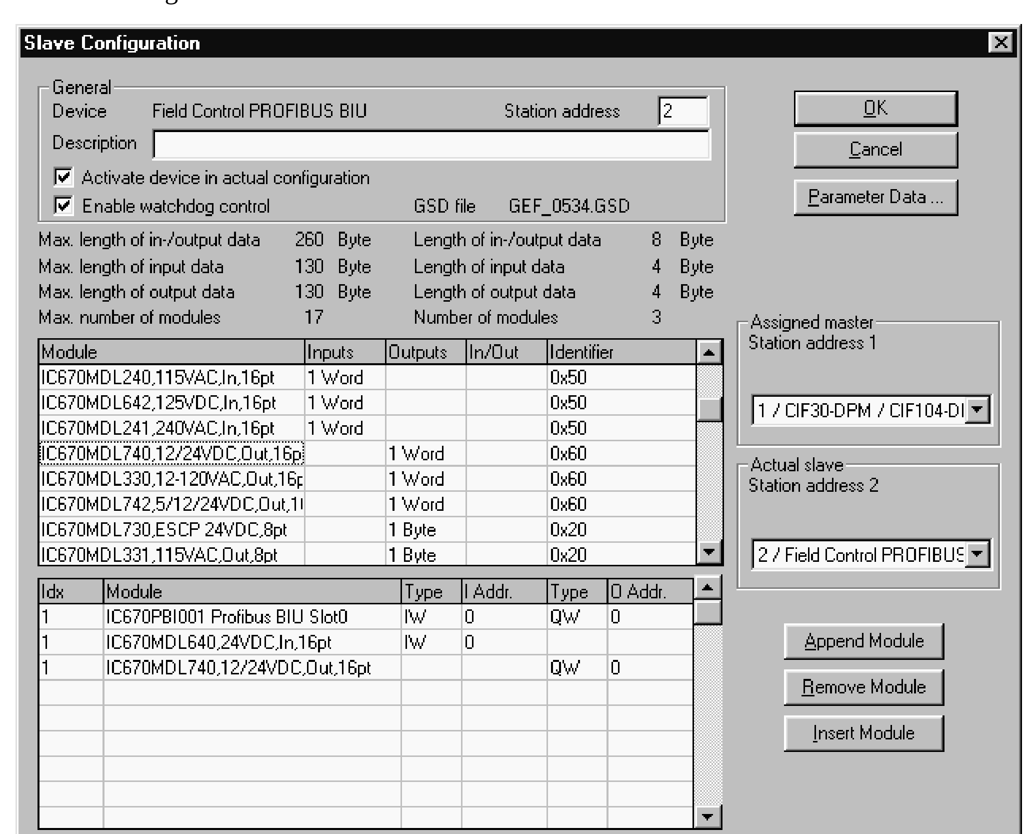


In diesem Beispiel ist der Slave eine modulare Field Control BIU. Zur Einstellung der E/A-Datenmenge, die mit dem Master ausgetauscht werden soll, müssen die einzelnen Module hinzugefügt werden. Bei Field Control müssen Sie zuerst die Profibus BIU Slot0 Konfiguration auswählen. Anschließend wählen Sie die einzelnen E/A-Module in der Reihenfolge der Steckplätze.

Hinweis

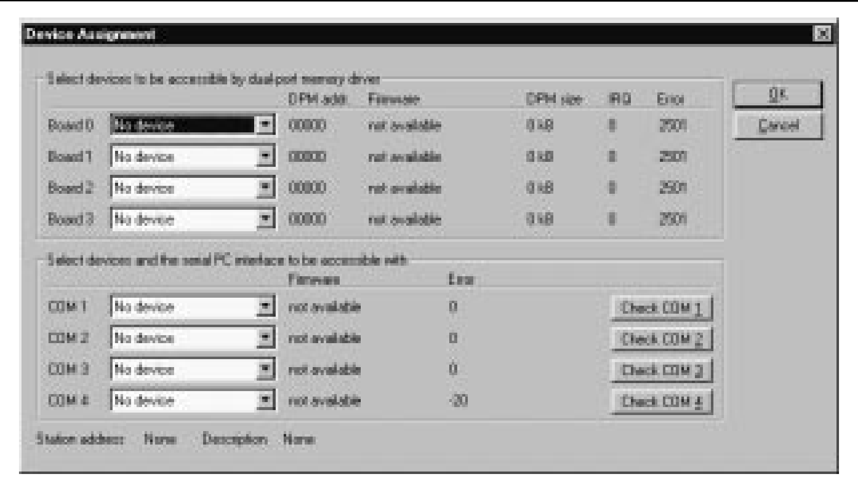
Bei intelligenten Modulen sehen Sie für jedes Modul zwei Einträge. Verwenden Sie die automatische Konfigurationsfunktion von Field Control, dann müssen Sie die Status/Steuerungskonfiguration auswählen, ehe Sie den E/A-Teil der Konfiguration auswählen.

Jetzt sollte am Bildschirm eine Dialogbox ähnlich der nachstehenden Abbildung zu sehen sein.

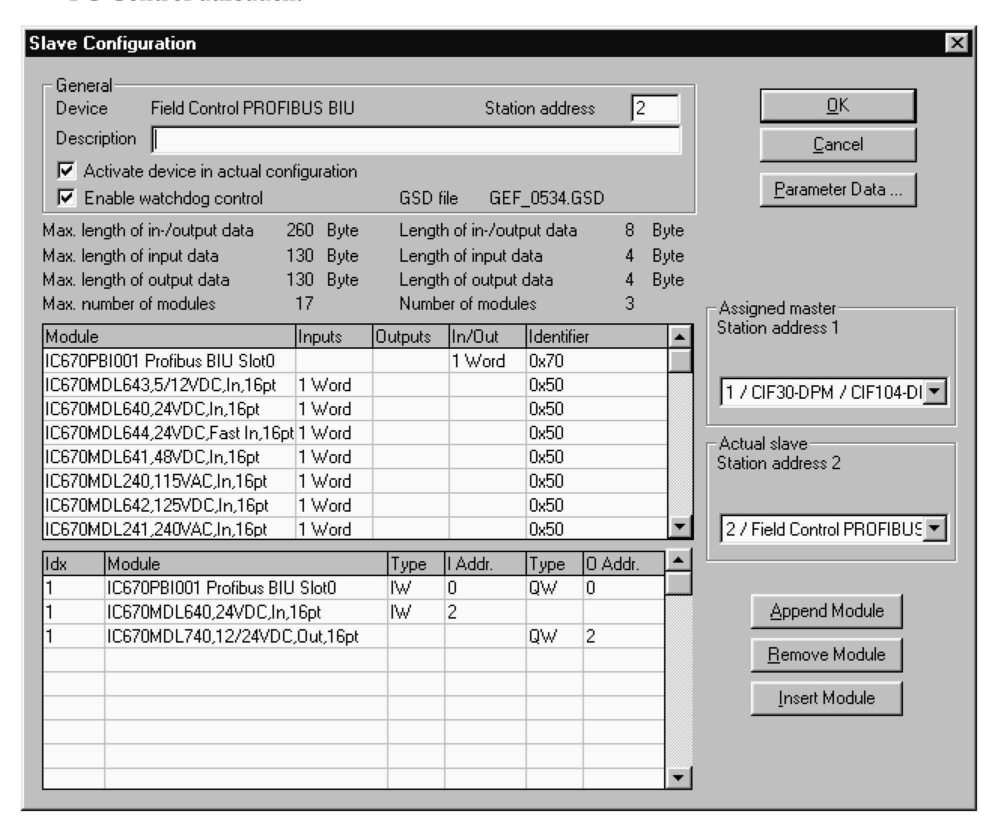


8. Drücken Sie die Taste OK, um zum Haupt-Netzwerkmenü zurückzukehren.

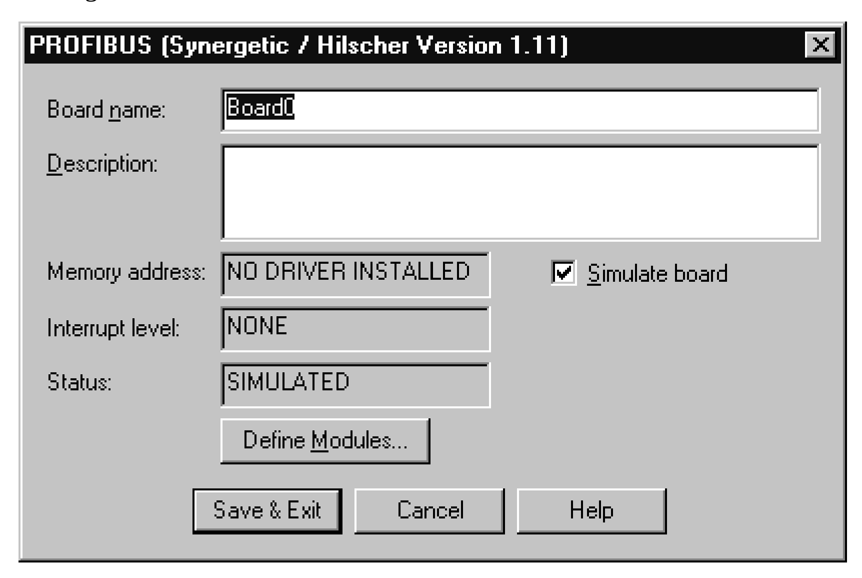
9. Um die erzeugte Konfiguration in das Mastermodul zu laden, wählen Sie aus dem Menü Online die Option "Herunterladen". Hierauf erscheint die Dialogbox "Gerätezuweisung". Wählen Sie die Pull-Down-Liste von Modul 0 und wählen das Menü CIF30-DPM aus. Klicken Sie auf OK, um das Herunterladen zu starten.



10. Nachdem das Herunterladen beendet ist, schauen Sie die Slave-Konfiguration der einzelnen Slave-Module an und notieren für jedes E/A-Modul die Werte von "I Addr" und "Q Addr". Im nachstehenden Menü haben diese Adressen den Wert 2. Sie benötigen diese Information, wenn Sie die Konfiguration unter PC Control aufbauen.

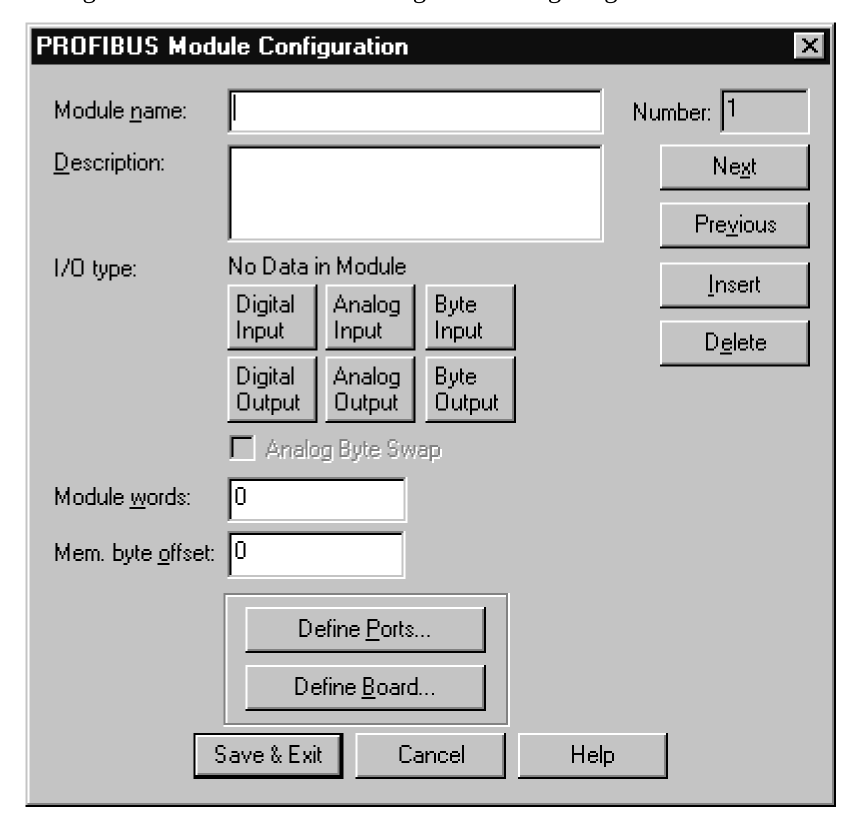


11. Starten Sie die PC Control Anwendung, Programmeditor.
12. Erzeugen Sie eine neue Konfiguration (wählen Sie "Neue Konfiguration" aus dem Pull-Down-Menü "Datei").
13. Konfigurieren Sie im ersten leeren Steckplatz ein Profibus-DP (Hilscher CIF30-DPM) Modul.
14. Um die E/A für das an dieser Karte angeschlossene Netzwerk zu konfigurieren, klicken Sie auf die Schaltfläche Moduldefinition. Hierauf erscheint die folgende Dialogbox.



Sie können jetzt dem Modul einen bezeichnenden Namen geben oder den angezeigten Standardwert belassen.

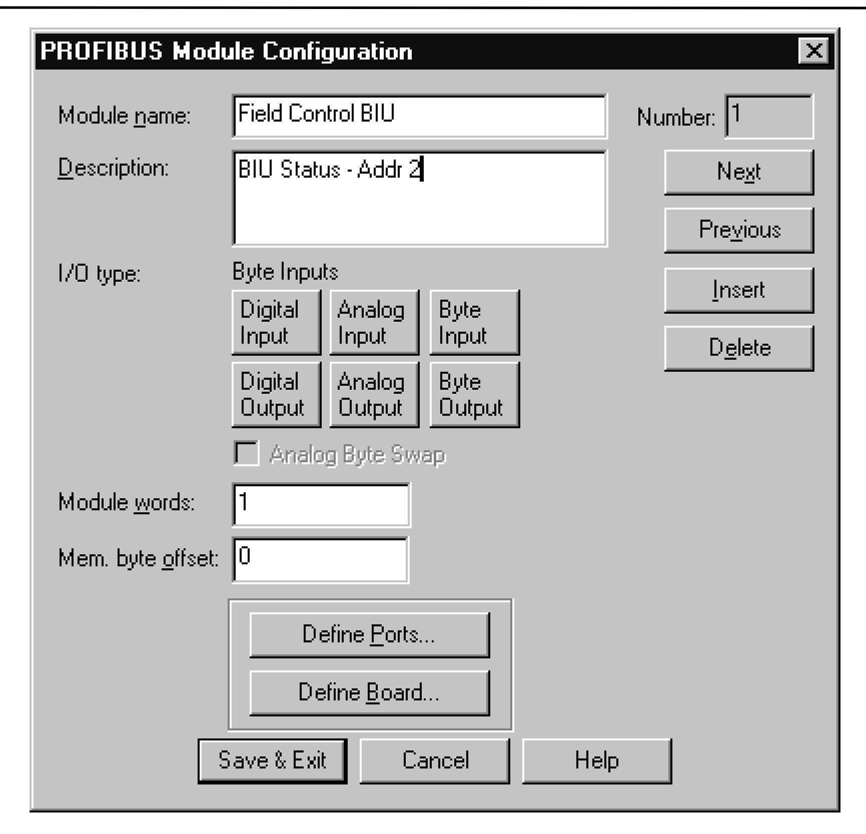
15. Um die Abbildung der Netzwerk-E/A-Daten auf die Variablen von PC Control zu definieren, wählen Sie die Schaltfläche "Moduldefinition". Hierauf wird die Dialogbox "PROFIBUS Modulkonfiguration" angezeigt.



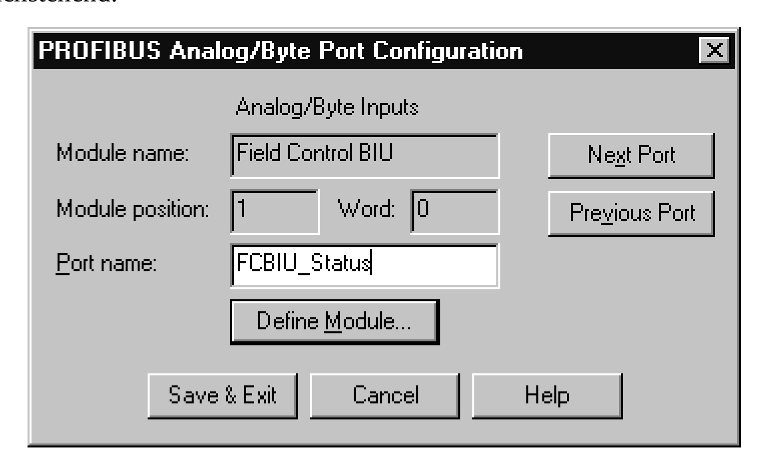
Für jedes von/zum Master gesendete/empfangene E/A-Element müssen Sie eine Modulkonfiguration erstellen. Definieren Sie zunächst für jedes Datenelement einen Namen und eine Beschreibung. Anschließend definieren Sie Datentyp, Länge und Offset in dem Master-Kartenspeicher (dies ist der Offset, den Sie zuvor notiert hatten). Nachstehend finden Sie Menübeispiele für eine Field Control BIU mit einem 16-Punkt-Eingangsmodul und einem 16-Punkt-Ausgangsmodul.

Hinweis

Field Control BIU Status/Steuerung und diskrete Module mit 16 Punkten sind als Byteeingang oder Byteausgang mit Modulworten gleich 1 definiert.



16. Für jedes definierte Datenelement müssen Sie auch einen eindeutigen Portnamen angeben. Nachdem Sie die Modulkonfiguration definiert haben, drücken Sie die Schaltfläche "Port definieren". Sie können jetzt dem Port einen beliebigen Namen geben oder den angezeigten Standardwert belassen. Da Sie diesen Namen zum Ansprechen der Daten in Ihrem Programm verwenden werden, sollten Sie möglichst einen Namen mit einer logischen Bedeutung wählen. Das Menü zur Definition des Ports für das BIU-Statuswort sehen Sie nachstehend:

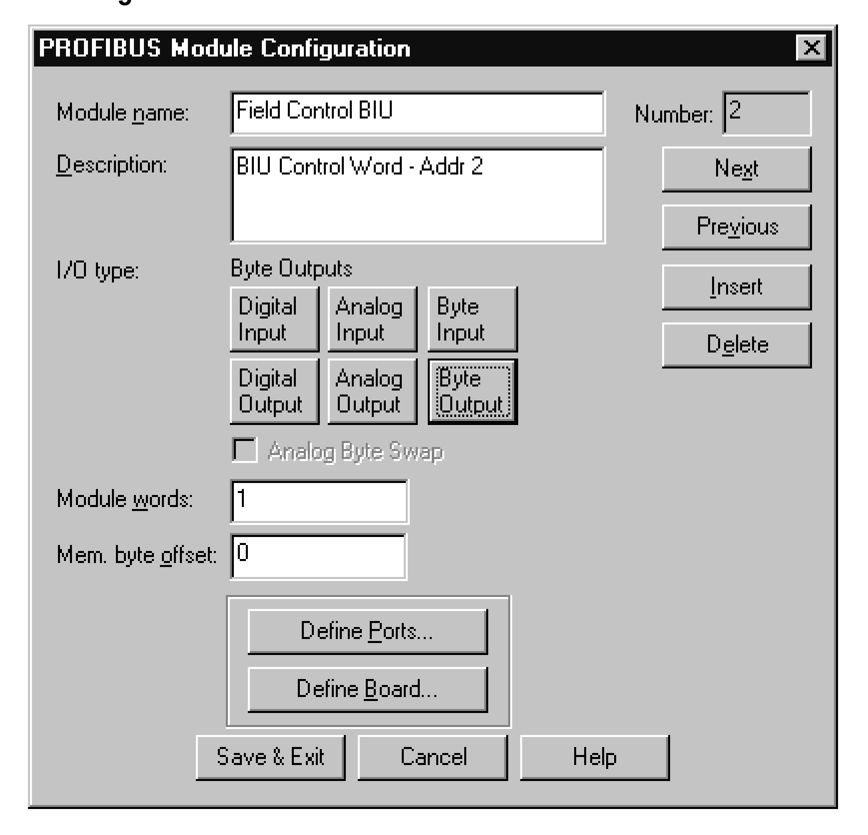


Hinweis

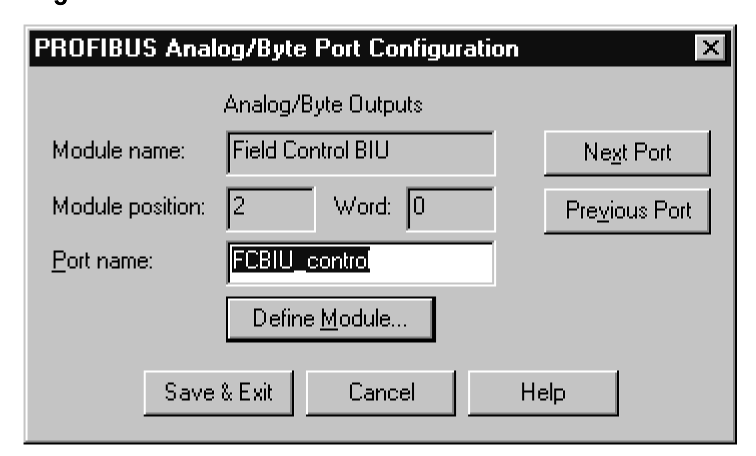
Bei Modulen mit mehreren Datenworten ist für jedes Teil eine Portdefinition erforderlich. Ein Beispiel hierfür ist ein 8-Kanal-Analogmodul. Dort sind 8 Portdefinitionen erforderlich. Sie können mit der Taste "Nächster Port" durch die Ports gehen.

Nachstehend sehen Sie Beispielmodule und Port-Konfigurationsmenüs für die restlichen BIU- und Moduldaten.

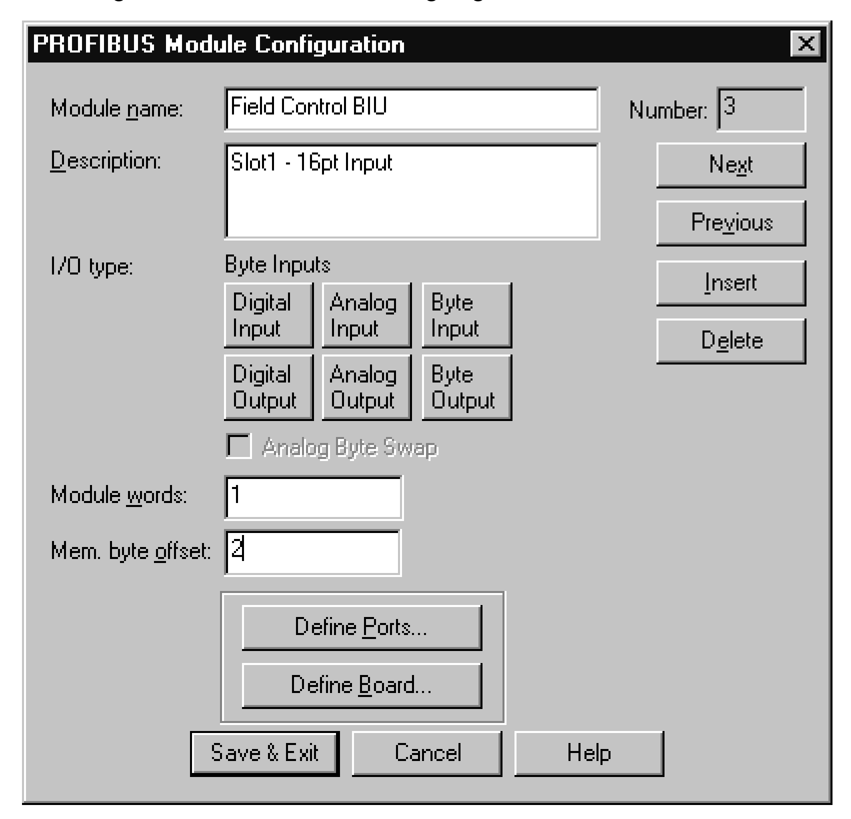
Modulkonfiguration für BIU Steuerwort



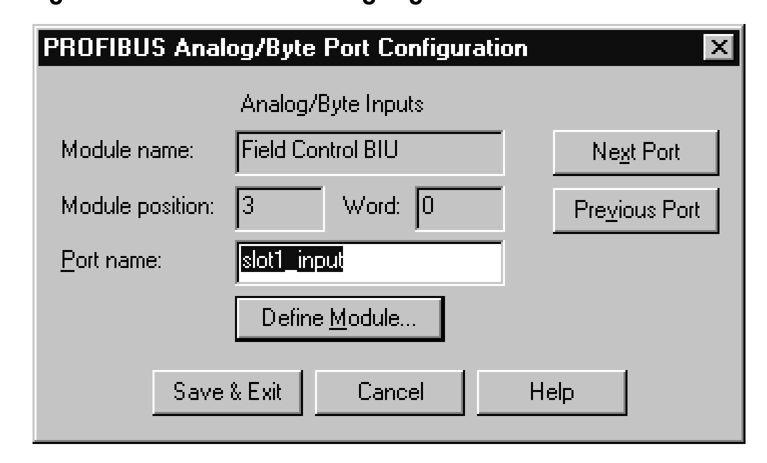
Port-Konfiguration für Steuerwort



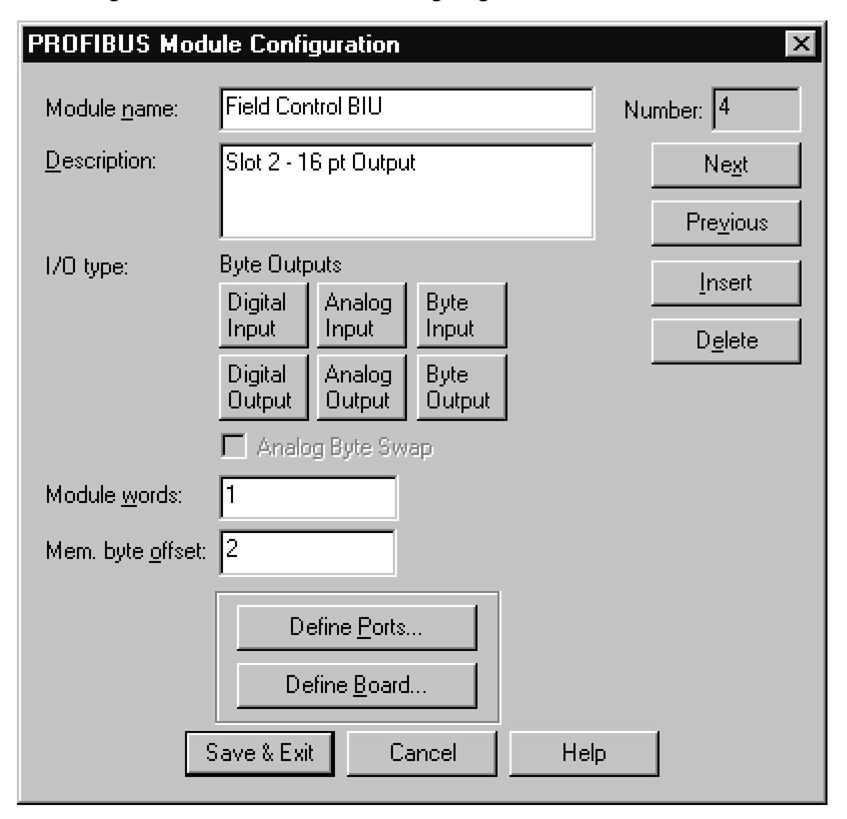
Modulkonfiguration für 16-Punkt Eingang



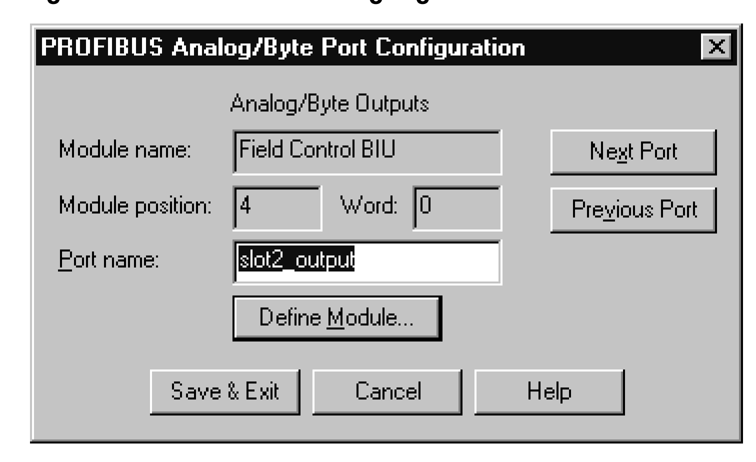
Portkonfiguration für 16-Punkt Eingang



Modulkonfiguration für 16-Punkt Ausgang

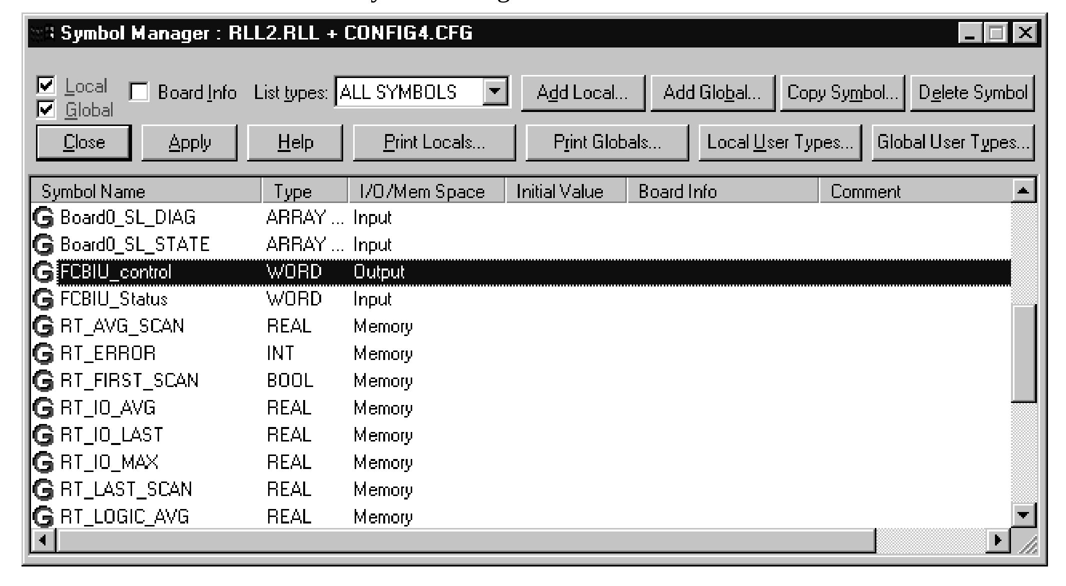


Portkonfiguration für 16-Punkt Ausgang

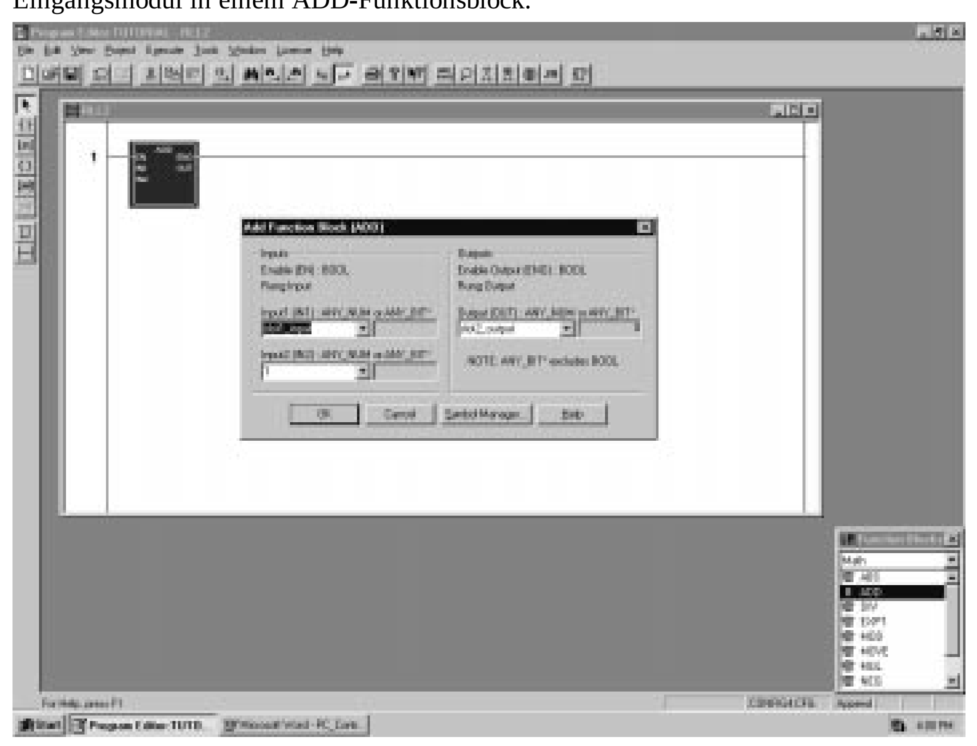


17. Im letzten Schritt bei der Konfigurationseinstellung wird die Konfiguration gespeichert und aktiviert.

Wenn Sie jetzt Ihren Programmeditor aufrufen, können Sie auf die Ports zugreifen, die Sie über den Symbolmanager definiert haben.



Das nachstehende Beispiel benutzt das in Steckplatz 1 eingebaute 16-Punkt Eingangsmodul in einem ADD-Funktionsblock.



Abschnitt 5: Dynamischer Datenaustausch

DDE-Schnittstelle

Die PC Control Software ermöglicht durch eine DDE-Schnittstellenoption zu Software anderer Hersteller den Datenaustausch mit der Steuerungssystemsoftware über eine DDE-Schnittstelle. Ist die DDE-Schnittstelle freigegeben, akzeptiert PC Control externe Programmbefehle und Schreib-/Leseanforderungen zu/von der Steuerungssystem-E/A und globalen Speichervariablen.

DDE-Kommunikation mit Microsoft Excel

Mit der DDE-Funktion in Microsoft Excel können Sie Daten zu und von globalen Symbolen übertragen. Mit Excel können Sie Daten vom Steuerungssystem zur Zusammenfassung und Analyse übertragen. Sie können auch Daten von Excel nach PC Control übertragen und damit die Programmausführung des Steuerungssystems kontrollieren oder Funktionen wie zum Beispiel ein maschinennahes Rezeptverwaltungssystem zur Verfügung stellen.

Daten zu Excel übertragen

Um den Wert eines globalen Symbols zu einer Zelle in einem Excel Tabellenblatt zu übertragen geben Sie in die Zelle, in der der Wert stehen soll, folgende Formel ein:

```
=ProgMgr|'_main _main'!VariableName
```

oder

```
=ProgMgr|'_main _main'!'ArrayName[4]'
```

Die Zeichenfolge '_main _main' ist das DDE-Thema mit dem Sie Globalvariablen holen können. Wie Sie aus der vorstehenden Formel sehen können, darf zwischen den beiden "_main" nur ein einziges Leerzeichen stehen. Ersetzen Sie VariableName mit dem Namen der Globalvariablen, die Sie übertragen wollen. Hierbei muß der Eintrag von VariableName die gleiche Groß- und Kleinschreibung wie in der Steuerungssystem-Anwendung aufweisen. Zum Zugriff auf Feldelemente verwenden Sie das zweite Format, mit dem Sie des entsprechende Element in jedem beliebigen Feld ansprechen können. Wegen der eckigen Klammern ([ ]) muß der Elementname in einzelne Anführungszeichen gestellt werden.

Diese Formel stellt eine automatische DDE-Verknüpfung zum Steuerungssystem her, über die der Wert im Excel-Tabellenblatt bei jeder Änderung der Variablen im Steuerungssystem aktualisiert wird. Ein ausführliches Beispiel finden Sie in der Online-Hilfe unter "Daten nach Excel übertragen” . Weitere Informationen zur Eingabe von Formeln in ein Excel-Tabellenblatt finden Sie in der Excel-Anwenderdokumentation.

Daten zum Steuerungssystem übertragen

Um Daten von Excel zu einer Globalvariablen im Steuerungssystem zu übertragen, benötigen Sie folgende in Excel Makrosprache kodierte DDE-Transaktionsroutine:

```
Sub transfer()

    Dim x

    x = Application.DDEInitiate("ProgMgr", "_main _main")

    Set rangeToPoke = Sheets("Sheet1").Cells(7, 11)

    Application.DDEPoke x, "VariableName", rangeToPoke

    Application.DDETerminate x

 End Sub
```

Wird diese Routine ausgeführt, dann überträgt sie den Wert der Zelle in Reihe 7 und Spalte 11 (K7) zu er Globalvariablen mit dem Namen VariableName. Wie Sie aus der vorstehenden Formel sehen können, darf zwischen den beiden "_main" nur ein einziges Leerzeichen stehen. Ersetzen Sie Sheet1 durch den Namen Ihrer Tabelle, die den zu übertragenden Wert enthält.

Werte auf Anforderung zum Steuerungssystem übertragen

Um Daten von Excel auf eine Anforderung vom Steuerungssystem hin zum Steuerungssystem zu übertragen, benötigen Sie folgende in Excel Makrosprache kodierte DDE-Transaktionsroutine:

```
Dim TimeSet As Double

Sub RunMeFirst()

    TimeSet = Now + TimeValue("00:00:05")

    Application.OnTime TimeSet, "Transfer"

End Sub

Sub Transfer()

    Dim x, y

    Dim z As Variant

    x = DDEInitiate("ProgMgr", "_main _main")

    z = DDERequest(x, "Trans1")

    y = Val(z(1))

    If y = 1 Then

        Set rangeToPoke = Sheets("Sheet1").Cells(2, 3)

        DDEPoke x, "Data1", rangeToPoke

        Set rangeToPoke = Sheets("Sheet1").Cells(3, 3)

        DDEPoke x, "Data2", rangeToPoke

        Set rangeToPoke = Sheets("Sheet1").Cells(4, 3)

        DDEPoke x, "Data3", rangeToPoke

        Set rangeToPoke = Sheets("Sheet1").Cells(1, 3)

        DDEPoke x, "Trans1", rangeToPoke

    End If

    DDETerminate x

    RunMeFirst

End Sub
```

Wird dieses Makro ausgeführt, dann überprüft es die Globalvariable Trans1 alle 5 Sekunden. Wird Trans1 auf 1 gesetzt, dann wird der Wert aus der Zelle in Reihe 2, Spalte 3 (C2) zur Globalvariablen Data1 übertragen. Entsprechend werden C3 nach Data2 und C4 nach Data3 übertragen. Anschließend wird C1, das auf 0 voreingestellt war, nach Trans1 übertragen, wodurch die Datenanforderung rückgesetzt wird. Wie bereits erwähnt, darf zwischen den beiden "_main" nur ein einziges Leerzeichen stehen. Ersetzen Sie Sheet1 durch den Namen Ihrer Tabelle, die die zu übertragenden Werte enthält. Um die Zeitspanne von 5 Sekunden zu verändern, verändern Sie den Wert TimeValue in der Funktion RunMeFirst.

Abschnitt 6: Konfiguration importieren/exportieren

Die Import/Export-Funktionen besitzen folgende Leistungsmerkmale:

- Bei Treibern, die Ausschneiden/Einfügen nicht gestatten, können Sie die Konfiguration exportieren. Verwenden Sie zunächst einen Texteditor zum Ausschneiden/Einfügen der Konfiguration und importieren Sie dann die Konfiguration zurück in das Steuerungssystem.
- Offline-Bearbeitung der Konfiguration. Anstelle des Steuerungssystems können Sie auch einen Texteditor oder ein CSV-kompatibles Dienstprogramm (z.B. Microsoft Excel) zum Bearbeiten der Konfiguration verwenden. Sie können aber auch Ihre eigene dialoggesteuerte Konfigurationssoftware (mit VB, VC++ usw.) zur Erzeugung von CSV-Dateien schreiben. Diese CSV können Sie dann in das Steuerungssystem importieren.
- Sie können Ihre eigene Anwendung zur Automatisierung der Generierung kundenspezifischer Konfigurationen erstellen.

Für CSV-Export und -Import werden folgende Konfigurationsdatentypen unterstützt: Globale Symbole, Yaskawa E/A-Treiber, GE 90/30 Treiber, PCIM Treiber.

Die Konfiguration wird als eine CSV-Datei exportiert, die von einem Texteditor oder Tabellenkalkulationsprogramm gelesen werden kann.

Konfiguration zu einer CSV-Datei exportieren

1. Öffnen Sie die Konfiguration, die Sie exportieren wollen.
2. Wählen Sie im Menü *Datei* die Option *Konfiguration nach CSV exportieren*. Hierauf erscheint eine Dateiauswahl-Dialogbox.



3. Wählen Sie einen Dateinamen aus oder geben Sie einen über die Tastatur ein. Klicken Sie dann auf *Öffnen*. Hierauf erscheint die Dialogbox *Konfiguration nach CSV exportieren*.

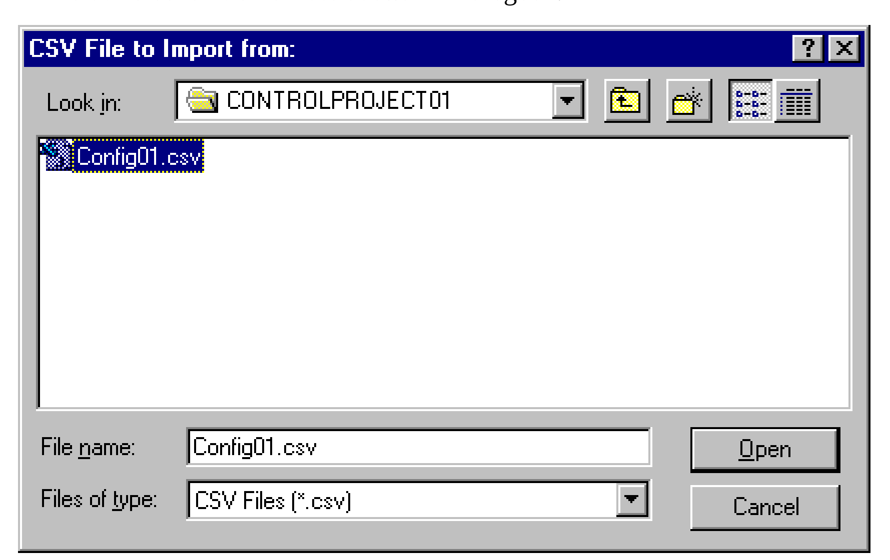


4. Stellen Sie die Exportoptionen nach Bedarf ein und klicken dann auf *OK*. Die Konfiguration wird in der angegebenen Datei gespeichert.

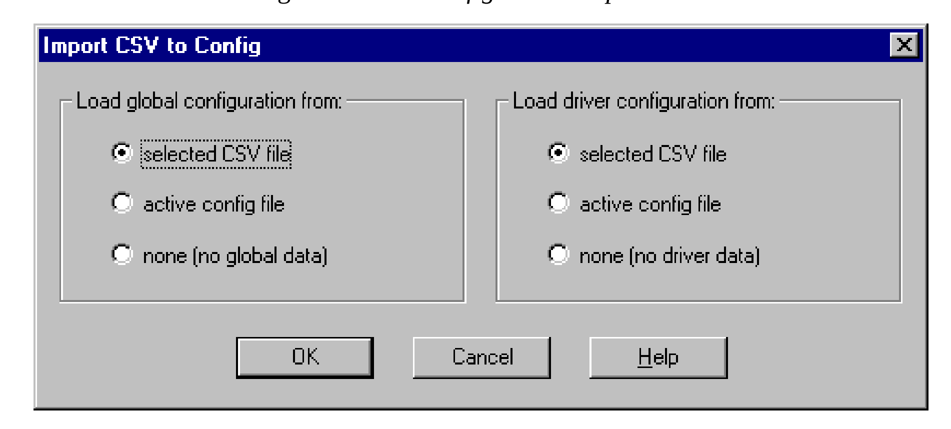
Eine CSV-Datei in eine Konfiguration importieren

1. Wählen Sie im Menü *Datei* die Option *CSV zu Konfiguration importieren*. Sie werden dann aufgefordert, die aktuell offene Konfiguration zu schließen.

2. Hierauf erscheint eine Dateiauswahl-Dialogbox.



3. Wählen Sie den Namen der zu importierenden Konfigurationsdatei aus oder geben Sie ihn über die Tastatur ein. Klicken Sie dann auf *Öffnen*. Hierauf erscheint die Dialogbox *CSV zu Konfiguration exportieren*.



4. Wählen Sie die Importoptionen nach Bedarf. Sie können globale Symbole und/oder Treiberdaten importieren, die globalen Symbole oder Treiberdaten aus der aktiven Konfiguration verwenden, oder die Konfiguration ohne globale Symbole oder Treiberdaten starten.

5. Klicken Sie dann auf *OK*. Die Konfigurationsdaten werden importiert und bilden nun die aktive Konfiguration.

Kapitel 4

Anwenderprogramme erstellen

Mit dem Programmeditor können Sie Anwenderprogramme erstellen, austesten, Syntaxanalyse durchführen und ausführen. Bei jedem Starten des Programmeditors werden Sie aufgefordert, ein Zugriffsebenen-Passwort einzugeben. Das Standard-Passwort für uneingeschränkten Zugriff lautet "**4567**".

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Konfiguration von Symbolen
- Strukturieren und Erzeugen von Kontaktplan-Anwenderprogrammen
- Strukturieren und Erzeugen von Ablaufplänen
- Erzeugen von Anwenderprogrammen in strukturiertem Text
- Erzeugen von Anweisungslisten-Anwenderprogrammen
- Bewegungsprogrammierung
- Online-Editieren

4

Abschnitt 1: Symbole konfigurieren

Über Symbole

Die Programmierwerkzeuge PC Control gestatten die Definition und Verwendung von Symbolen, bei denen es sich um interne Speicherplätze handelt,die Informationen enthalten. Die Informationen selbst werden vom Datentyp bestimmt; es können reelle Zahlen, ganze Zahlen, Zeichenfolgen usw. sein. Globale Symboldefinitionen werden in der aktiven Konfigurationsdatei gespeichert. Lokale Symboldefinitionen dagegen werden in der Anwenderprogrammdatei (Ablaufplan, Kontaktplan, strukturierter Text oder Anweisungsliste) gespeichert.

E/A-Punkte und System- und Laufzeitsymbole sind vordefiniert und können in Programmen zwar angezeigt und verwendet, aber nicht editiert werden. Sie erscheinen als globale Symbole.

Anwendungsbereich der Symbole

Der Anwendungsbereich eines Symbols kann bezüglich eines Programms lokal oder global sein.

Lokales Symbol

Ein lokales Symbol kann nur in dem Programm angesprochen werden, in dem es definiert wurde, oder in Makroschritten, die von dem Programm aufgerufen werden. Das Programm muß für Sie offen sein, so daß Sie die darin verwendeten lokalen Symbole definieren können. In der Bedienerschnittstelle für DDE-Operationen können Sie kein lokales Symbol aufrufen. Die Definitionen lokaler Symbole werden in der Anwenderprogrammdatei gespeichert.

Globales Symbol

Auf ein globales Symbol können alle Programme innerhalb eines Projekts zugreifen. In der Bedienerschnittstelle oder für DDE-Operationen können Sie globale Symbole verwenden. Globale Symboldefinitionen werden in der aktiven Konfigurationsdatei gespeichert.

E/A-Punkte

E/A-Punkte sind externe Adressen. Da Sie sie in einem Programm wie Symbole ansprechen können, erscheinen die E/A-Punkte im Symbolmanager und werden als globale Symbole aufgelistet.

E/A-Punkte funktionieren wie globale Symbole: Sie können Sie aus jedem beliebigen Programm heraus ansprechen und Sie können sie in der Bedienerschnittstelle oder bei DDE-Operationen verwenden. E/A-Symbole können nur im Konfigurations-Dienstprogramm editiert werden, nicht im Symbolmanager.

Funktionsblöcke

Die meisten Funktionsblöcke müssen instantiiert werden. Um in Programmen mit strukturiertem Text oder Anweisungslisten verwendet werden zu können, muß eine Instanz des Funktionsblocktyps explizit im Symbolmanager vereinbart werden.

Systemobjekte (PID, PRGBC und TMR)

Systemobjekte müssen instantiiert werden. Um in strukturiertem Text verwendet werden zu können, muß eine Instanz des Systemobjekttyps explizit im Symbolmanager vereinbart werden.

Bezeichner

Bezeichner werden als Symbolnamen verwendet. Sie besitzen folgende Eigenschaften:

- Sie können aus Groß- oder Kleinbuchstaben (A—Z und a—z), Ziffern (0—9) und Unterstrich (_) bestehen.
- Sie müssen mit einem Buchstaben oder einem einzelnen Unterstrich beginnen.
- Groß- und Kleinschreibung wird beachtet.
- Die Stellung der einzelnen Zeichen ist für die Einmaligkeit eines Bezeichners maßgebend.
- Mehrere aufeinanderfolgende Unterstriche sind nicht zulässig.
- Leerzeichen sind nicht zulässig.
- Ein Bezeichner kann maximal 100 Zeichen lang sein.
- Reservierte System-Schlüsselwörter dürfen nicht als Bezeichner verwendet werden. Siehe "Schlüsselwörter” für eine Liste der Schlüsselwörter.
- Muß in seinem Anwendungsbereich einmalig sein.

Beispiele zulässiger Bezeichner sind:

_Sym1	
Sym_Two_A	
SYM	(SYM und sym werden als unterschiedliche Bezeichner betrachtet)
Sym	
A	(A und AA werden als unterschiedliche Bezeichner betrachtet)
AA	

Beispiele unzulässiger Bezeichner sind:

1A	(beginnt mit einer Zahl)
Sym___Two	(mehrere aufeinanderfolgende Unterstriche)
Sym Two	(enthält ein Leerzeichen)
Sym&One	(enthält ein unzulässiges Zeichen)
END	(ein reservierter Bezeichner)

Literale

Literale werden häufig zur Definition oder Darstellung von Datenwerten. Literale können zum Beispiel benutzt werden als Eingangswert von Funktionen oder Funktionsblöcken, zur Zuweisung von Werten zu Variablen oder Konstanten sowie innerhalb von Programmanweisungen. Es gibt vier verschiedene Literaltypen: Numerisch, Zeichenfolge, Zeitdauer und Uhrzeit mit Datum.

Numerische Literale

Numerische Literale sind entweder ganze oder reelle Zahlen. Ganzzahlige Literale können dezimal, binär, oktal oder hexadezimal sein.

0	456	+34	-7_000	ganzzahlige Literale
0.0	0.11			reelle Literale
2#1010_1010 (170 dezimal)				binäres Literal
8#252 (170 dezimal)				oktales Literal
16#AA (170 dezimal)				hexadezimales Literal
FALSE	0	TRUE	1	Boolesches Literal

Numerische Literale besitzen folgende Eigenschaften:

- Ein numerisches Literal kann einzelne Unterstriche (_) enthalten, die den Wert des Literals nicht beeinflussen.
- Reelle Literale enthalten einen Dezimalpunkt.
- Vor dezimalen Literalen ist ein Vorzeichen (+ oder -) möglich.
- Die Schlüsselwörter FALSE und TRUE entsprechen den Booleschen Werten 0 bzw. 1.

Zeichenfolgen-Literale

Ein Zeichenfolgen-Literal ist eine Zeichenfolge aus 0 oder mehr Zeichen, die durch einzelne Anführungszeichen (') voneinander getrennt sind. Das Dollarzeichen $ findet in einem Zeichenfolgen-Literal eine besondere Verwendung. Folgen ihm zwei Hexadezimalstellen, wird es als die Hexadezimaldarstellung des 8-Bit-Zeichencodes interpretiert. In Zeichenfolgen aus zwei Zeichen wird es auch zur Darstellung des Dollarzeichens ($), einzelner Anführungszeichen (') und spezifizierter nicht abdruckbarer Zeichen verwendet.

Beispiele von Zeichenfolgen-Literalen:

''	leere Zeichenfolge
'XYZ'	Zeichenfolge aus drei Zeichen
' '	ein Leerzeichen
'$41 $42 $43'	eine Zeichenfolge aus fünf Zeichen: 'A B C'
'$$'	Dollarzeichen
'$' '	einzelnes Anführungszeichen
'$L' oder '$l'	Zeilenvorschub
'$N' oder '$n'	neue Zeile
'$P' oder '$p'	Formularvorschub
'$R' oder '$r'	Wagenrücklauf
'$T' oder ' $t'	Tabulator

Zeitdauer-Literale

Ein Zeitdauer-Literale beginnt mit dem Schlüsselwort T#, TIME#, t#, oder time#, dem eine oder mehrere Zeiteinheiten folgen. Beispiele von Zeitdauer-Literalen:

T#1D1H1M1S	1 Tag, 1 Stunde, 1 Minute, 1 Sekunde
Time#1d_1h_1m_1s	wie oben
time#25h1ms	25 Stunden, 1 Millisekunde
t#1m_2.5s	1 Minute, 2,5 Sekunden

Zeitdauer-Literale besitzen folgende Eigenschaften:

- Die Zeiteinheiten können in Groß- oder in Kleinbuchstaben geschrieben werden. D, d=Tage; H, h=Stunden; M, m=Minuten; S, s=Sekunden; und MS, ms=Millisekunden.
- Zeitdauer-Einheiten können durch einen Unterstrich (_) voneinander getrennt werden.
- Die höchstwertige Einheit eines Zeitdauer-Literals kann überlaufen.
- Die niedrigstwertige Einheit eines Zeitdauer-Literals kann als reelle Zahl (ohne Exponent) geschrieben werden.

Uhrzeit-mit-Datum-Literale

Am Anfang eines Uhrzeit-mit-Datum-Literals steht eines der folgenden Schlüsselwörter, dem dann Uhrzeit und Datum im geeigneten Format folgen.

DATE#YYYY-MM-DD	nur Datum
D#YYYY:MM:DD	wie oben
TIME_OF_DAY#HH:MM:SS.MS	nur Zeit
TOD#HH:MM:SS.MS	wie oben
DATE_AND_TIME#YYYY-MM-DD-HH:MM:SS.MS	Datum und Zeit
DT#YYYY-MM-DD-HH:MM:SS.MS	wie oben

Beispiele für Uhrzeit-mit-Datum-Literale:

```
DATE#1998-02-13
D#1998-02-13
TIME_OF_DAY#12:00:00
TOD#12:00:00.01
DATE_AND_TIME#1998-02-13-12:00:00.01
```

- Die Schlüsselwörter für Datum und Uhrzeit können abgekürzt werden. DATE oder D; TIME_OF_DAY oder TOD; DATE_AND_TIME oder DT.

Datentypen

Datentypen müssen Symbolen (Variablen und Konstanten) zugewiesen werden. Die Eigenschaften der elementaren Datentypen werden in den folgenden Abschnitten dargestellt. Generische Datentypen werden in "Generische Datentypen**”** beschrieben, anwenderdefinierte Datentypen in "Anwenderdefinierter Datentyp**”**.

In den nachstehenden Beschreibungen werden folgende Begriffe verwendet:

- Generischer Typ gibt die generischen Typen an, für die der beschriebene Datentyp eingesetzt werden kann.
- Größe gibt an, wieviel Speicherplatz eine Instanz des betreffenden Datentyps belegt (der 1-Bit Boolesche Typ wird zum Beispiel in Wirklichkeit als ein Byte abgelegt).
- Bereich gibt an, in welchem Wertebereich eine Instanz des Datentyps liegen kann.
- Standardwert ist der ursprüngliche Standardwert, der einer Instanz des betreffenden Datentyps zugewiesen wird, wenn sonst kein anderer Anfangswert angegeben wurde.

BOOL (Boolescher Wert)

Ein BOOL kann einen von zwei Zuständen annehmen: 0 oder 1, entsprechend FALSE oder TRUE.

Generischer Typ	ANY, ANY_BIT.
Größe	1 Bit
Bereich	0 (FALSE), 1 (TRUE).
Standardwert	0

BYTE

Ein BYTE ist eine Bitfolge der Länge 8.

Generischer Typ	ANY, ANY_BIT.
Größe	1 Byte
Bereich	Nicht anwendbar
Format	MSB 7 … 0 LSB
Standardwert	00000000

DATE

Mit diesem Datentyp wird ein Datum (ausschließlich) im Format JJJJ-MM-TT dargestellt.

Beim Erstellen eines Ausdrucks von DATE Datentypen müssen alle Werte vom gleichen Typ sein und das Ergebnis muß ein Datum sein.

Generischer Typ	ANY, ANY_DATE.
Größe	4 Bytes
Bereich	-
Standardwert	D#0001-01-01

DINT (doppeltgenaue ganze Zahl)

Der Datentyp DINT stellt eine ganze Zahl mit Vorzeichen dar, die aus einer oder mehreren Ziffern (0—9) besteht und keinen Dezimalpunkt enthalten darf.

Generischer Typ	ANY, ANY_NUM, ANY_INT.
Größe	4 Bytes.
Bereich	-2147483648 bis +2147483647.
Standardwert	0

DWORD (Doppelwort)

Ein DWORD ist eine Bitfolge der Länge 32.

Generischer Typ	ANY, ANY_BIT.
Größe	4 Bytes.
Bereich	Nicht anwendbar
Format	

MSB 31 ... LSB 0

Standardwert	0

INT (ganze Zahl)

Der Datentyp INT stellt eine ganze Zahl mit Vorzeichen dar, die aus einer oder mehreren Ziffern (0—9) besteht und keinen Dezimalpunkt enthalten darf.

Generischer Typ	ANY, ANY_NUM, ANY_INT.
Größe	In einer Erweiterung der IEC 1131-3 Spezifikation umfaßt der INT 4 Bytes.
Bereich	-2147483648 bis +2147483647.
Standardwert	0

REAL

Der Datentyp REAL ist ein vorzeichenbehafteter 64-Bit-Wert, der aus einer oder mehreren Ziffern (0—9) besteht und einen Dezimalpunkt enthält.

Generischer Typ	ANY, ANY_NUM, ANY_REAL.
Größe	8 Bytes.
Bereich	-3.402823 E38 bis -1.401298 E-45 (negativ), +1.401298 E-45 bis +3.402823 E38 (positiv).
Standardwert	0.0

STRING

ASCII-Zeichenfolge variabler Länge.

Generischer Typ	ANY.
Größe	64 Bytes.
Standardwert	'' (leere Zeichenfolge)
Format	Folge von ASCII-Zeichen in einzelnen Anführungszeichen. Beispiel: ' Dies ist eine zulaessige Zeichenfolge. '

TIME

Mit diesem Datentyp können Sie eine Zeitdauer im Format T#[nD][nH][nM][nS][nMS] darstellen, wobei n die Anzahl Tage, Stunden, Minuten, Sekunden und Millisekunden angibt.

Beim Erstellen eines Ausdrucks von TIME Datentypen müssen alle Werte vom gleichen Typ sein und das Ergebnis muß eine Zeit sein.

Generischer Typ	ANY.
Größe	-
Bereich	-
Standardwert	T#0S
Format	

TOD (TIME_OF_DAY)

Mit diesem Datentyp wird eine Uhrzeit (ausschließlich) im Format HH:MM:SS dargestellt.

Beim Erstellen eines Ausdrucks von TOD Datentypen müssen alle Werte vom gleichen Typ sein und das Ergebnis muß eine Uhrzeit sein.

Generischer Typ	ANY, ANY_DATE.
Größe	4 Bytes.
Bereich	00:00:00 bis 23:59:59
Standardwert	TOD#00:00:00
Format	HH:MM:SS (Stunden:Minuten:Sekunden).

UINT (ganze Zahl ohne Vorzeichen)

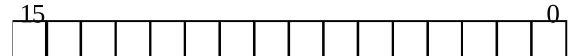
Der Datentyp UINT stellt eine ganze Zahl ohne Vorzeichen dar, die aus einer oder mehreren Ziffern (0—9) besteht und keinen Dezimalpunkt enthalten darf.

Generischer Typ	ANY, ANY_NUM, ANY_INT.
Größe	2 Bytes.
Bereich	0 bis 65535.
Standardwert	0

WORD

Ein WORD ist eine Bitfolge der Länge 16.

Generischer Typ	ANY, ANY_BIT.
Größe	2 Bytes.
Bereich	Nicht anwendbar
Format	MSB 15 … LSB 0



Standardwert	0

Generische Datentypen

Die nachstehende Tabelle zeigt die Datentypen und die Hierarchie der generischen Typen. Die generischen Typen sind diejenigen, vor denen ein ANY_ steht. Sie werden entsprechend ihrer Eignung in den Funktions- oder Funktionsblockbeschreibungen verwendet (anstelle ausführlicher Angaben langer Listen von Datentypen).

Läßt zum Beispiel ein Funktionsblock als Eingabewert einen Datentyp ANY_NUM zu, dann kann dem Funktionseingang ein Symbol zugewiesen werden, das vom Datentyp REAL, einem ganzzahligen Typ (INT, DINT, usw.), oder einem Bitfolgetyp (WORD, DWORD, usw.) ist. Läßt ein Funktionsblock als Eingabewert einen Datentyp ANY_BIT zu, dann kann dem Funktionseingang nur ein Symbol vom Datentyp WORD, DWORD, usw. zugewiesen werden.

Datentyphierarchie					
			ANY_REAL	REAL	
				ANY_INT	DINT, INT, UINT
				ANY_BIT	DWORD, WORD, BYTE, BOOL
		STRING			
		ANY_DATE			DATE, TIME_OF_DAY
		TIME			
	Abgeleitet				

Anwenderdefinierter Datentyp

Für komplexere Datenverarbeitung können Sie selbststrukturierte Datentypen erstellen. Eine Struktur kann mehrere Glieder aus unterschiedlichen Grundtypen oder anwenderdefinierten Strukturtypen enthalten. Betrachten wir zum Beispiel eine anwenderdefinierte Struktur mit der Bezeichnung UserStructure01, die aus einem ganzzahligen Glied USInt, einem Booleschen Typ USBool und einem Zeichenfolgetyp USString besteht. Auf die einzelnen Glieder kann auf folgende Weise zugegriffen werden:

```
UserStructure01.USInt:=101;
UserStructure01.USBool:= TRUE;
UserStructure01.USString:="ABC";
```

Anwenderdefiniert Typen können überall dort eingesetzt werden, wo ein Datentyp ANY oder USER-DEFINED zugelassen ist. Weitere Angaben finden Sie unter "Anwenderdefinierte Datentypen editieren”.

Felder

Um auf ein bestimmtes Element eines Feldes zuzugreifen geben Sie den Symbolnamen ein, gefolgt von eckigen Klammern, in die Sie die Nummer des gewünschten Elements eintragen. Um zum Beispiel auf das fünfte Element eines Feldes mit dem Namen Myarray zuzugreifen, müssen Sie Myarray[5] eingeben. Hierbei wird angenommen, daß Sie die Untergrenze des Feldes auf 1 eingestellt haben. Sie können ein Feld auch indizieren, indem Sie in die eckigen Klammern einen Symbolnamen vom Typ INT eintragen.

Pointersymbole

Vorsicht

Pointer sollten nur von Fachleuten eingesetzt werden. Falsche Verwendung kann zu unvorhersehbarem Betriebsverhalten und großen Problemen beim Austesten führen.

Strukturierter Text besitzt zwei Pointeroperatoren: Den Pointer Referenzierung & Operator und den Pointer Dereferenzierung * Operator. Diese Operatoren werden zur indirekten Adressierung eingesetzt. Erfahrene Anwender finden weitere Informationen hierzu in Anhang E.

Symbolmanager

Mit dem Symbolmanager können Sie Variablen und Konstantensymbole sowie Funktionsblockinstanzen anzeigen, erstellen, editieren, kopieren oder löschen. E/A-Punkte und Systemsymbole können nur angezeigt werden. Der Symbolmanager zeigt eine Liste der in der aktiven Datei definierten Symbole und der in der aktiven Konfiguration definierten globalen Symbole an.

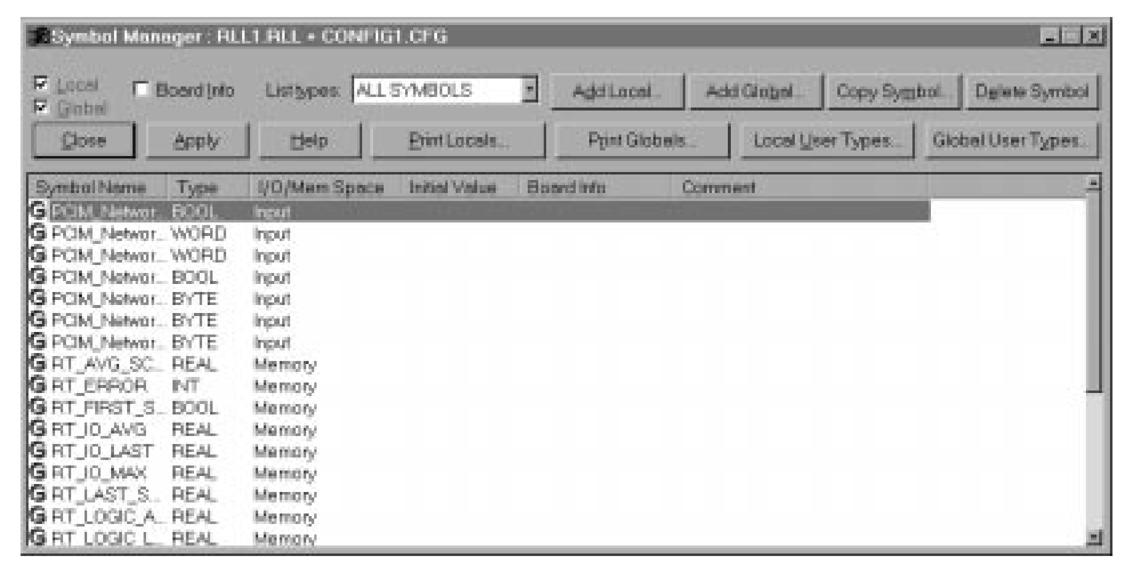
Mit Drag&Drop können Sie im Symbolmanager Symbole zu Programmanweisung im strukturierten Text , Anweisungslisten-Programmanweisung, oder Kontakte und Spulen in Kontaktplanprogrammen hinzufügen. Mit Drag&Drop können Sie auch Symboldaten zu anderen Softwareprogrammen mit OLE-Fähigkeit (z.B. Microsoft WordPad, Word, Excel oder ähnliche) übertragen. Sie können auch Symbole aus den Listen in den verschiedenen Dialogboxen auswählen.

Symbolmanager öffnen

Zum Öffnen des Symbolmanagers wählen Sie im Menü *Werkzeuge* des Programmeditors oder Bedienerschnittstellen-Editors die Option Symbolmanager.

- Der Symbolmanager kann auch über die einzelnen Dialogboxen in den Editoren geöffnet werden.
- Der Symbolmanager kann nicht gleichzeitig im Programmeditor und im Bedienerschnittstellen-Editor geöffnet sein.

Die nachstehende Abbildung zeigt ein Beispiel des Symbolmanagers. Die Beschreibung der E/A- und Speichersymbole wird tabellarisch angezeigt. Vor den einzelnen Symbolnamen steht entweder ein G für globale Symbole oder ein L für lokale Symbole. Ein Sternchen (*) vor dem Symbolnamen zeigt an, daß es sich um ein Pointersymbol handelt. Gehört der Symbolnamen zu einem E/A-Punkt, dann wird die Platineninformation in der entsprechenden Spalte angezeigt.



Bedienelemente des Symbolmanagers

Bedienelement	Beschreibung
Lokal Global	Enthält dieses Kästchen einen Haken, dann wird der entsprechende Symboltyp in der Symbolliste angezeigt.
Moduldaten	Enthält dieses Kästchen einen Haken, dann werden die Moduldaten (Schnittstellenplatine, Chassis, usw.) angezeigt.
Listentypen	Filtert die Symbolliste nach Datentyp und Funktionsblocktyp.
Lokal hinzu	Ein lokales Symbol erstellen und hinzufügen.
Global hinzu	Ein globales Symbol erstellen und hinzufügen.
Symbol kopieren	Kopiert das aktuell ausgewählte Symbol.
Symbol löschen	Löscht das aktuell ausgewählte Symbol.
Schließen	Schließt den Symbolmanager.
Anwenden	Macht die Symboländerungen außerhalb des Symbolmanagers sichtbar.
Hilfe	Zeigt Hilfe zum Symbolmanager an.
Lokale drucken	Druckt eine Liste der lokalen Symbole.
Globale drucken	Druckt eine Liste der globalen Symbole.
Lokale Anwendertypen	Erstellt einen lokalen Anwendertyp.
Globale Anwendertypen	Erstellt einen lokalen Anwendertyp.

Symbol erstellen

Um ein neues Symbol zu erstellen klicken Sie entweder auf die Taste *Lokal hinzu* oder *Global hinzu*, oder benutzen das Kontextmenü in der Symbolliste. Hierauf erscheint die Dialogbox *Symboldetails*. Geben Sie die entsprechenden Details in der Dialogbox *Symboldetails* ein und klicken dann auf *OK*. Das neue Symbol erscheint in der Symbolliste.

Ist kein Programm geöffnet, können Sie nur globale Symbole erstellen. Ist ein Programm offen, können Sie lokale und globale Symbole erstellen. Ein lokales Symbol kann nur von dem Programm benutzt werden, das bei der Definition des Symbols offen (aktiv) war.

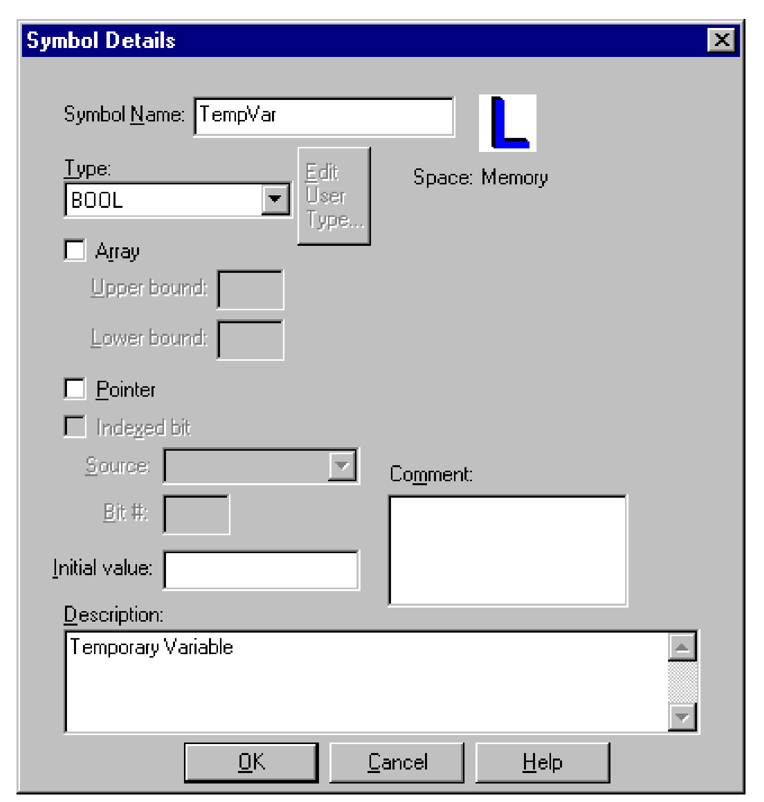
Hinweis

Nach dem Editieren von Symbolen müssen Sie immer auf *Anwenden* klicken, damit die Änderungen außerhalb des Symbolmanagers sichtbar werden.

Symbol editieren

1. Wählen Sie das Symbol aus, das Sie editieren wollen, und drücken dann Enter, oder doppelklicken Sie auf das Symbol.
2. Hierauf erscheint die Dialogbox *Symboldetails.*
3. Bearbeiten Sie die Daten und klicken Sie auf *OK*, um die Änderungen zu übernehmen.

Symboldetails



Feld	Beschreibung
Symbolname	Als Symbolname kann jeder gültige Bezeichner verwendet werden.
Typ	Der elementare oder anwenderdefinierte Strukturdatentyp oder Funktiosblocktyp des Symbols. Weitere Angaben finden Sie unter"Datentypen”.
Anwendertyp editieren	Wurde das Symbol als Anwendertyp definiert, dann ist diese Schaltfläche aktiv und der Anwendertyp kann bearbeitet werden. Weitere Angaben finden Sie unter "Anwenderdefinierte Datentypen editieren**”**.
Feld	Enthält dieses Kästchen einen Haken, dann wird das Symbol als Feld definiert. Geben Sie die obere und untere Feldgrenze an. Weitere Angaben finden Sie unter Felder.
Pointer	Enthält dieses Kästchen einen Haken, dann wird das Symbol als Pointer definiert. Weitere Angaben finden Sie unter Pointersymbole.
Indiziertes Bit	Über die Eigenschaft "indiziertes Bit" können Sie ein spezifisches Bit innerhalb eines Byte-, Wort- oder Doppelwortsymbols ansprechen. Weitere Angaben finden Sie unter Ein Bit in einem Symbol benennen.
Anfangswert	Hiermit können Sie dem Symbol einen Anfangswert zuweisen. Bei jedem Programmlauf wird ein lokales Symbol auf seinen Anfangswert zurückgesetzt. Ein globales Symbol erhält seinen Anfangswert zugewiesen, wenn die Konfiguration aktiviert wird.
Kommentar	Eine kurze Beschreibung, die innerhalb des Symbolmanagers erscheint.
Beschreibung	Die Beschreibung erscheint nur in der Symboldetail-Dialogbox, nicht im Symbolmanager.

Symbol kopieren

Mit dem Kopierbefehl können Sie schnell ein neues Symbol erstellen, das die gleichen Eigenschaften wie ein bestehendes Symbol besitzt.

1. Wählen Sie das Symbol aus, das Sie kopieren wollen.
2. Klicken Sie auf die Schaltfläche *Symbol kopieren* oder verwenden Sie das Kontextmenü. Die Dialogbox *Symboldetails* öffnet sich. Alle Parameter sind genau wie bei dem ausgewählten Symbol eingestellt.
3. Sie müssen einen neuen Symbolnamen eingeben. Geben Sie den neuen Symbolnamen ein, führen dann alle weiteren erforderlichen Änderungen durch und klicken dann auf *OK*.

Hinweis

Nach dem Editieren von Symbolen müssen Sie immer auf *Anwenden* klicken, damit die Änderungen außerhalb des Symbolmanagers sichtbar werden.

Ein Symbol löschen

Wählen Sie das Symbol aus, das Sie löschen wollen. Klicken Sie dann auf die Schaltfläche *Symbol löschen* oder verwenden Sie das Kontextmenü.

Ein Bit in einem Symbol benennen

Über die Eigenschaft "indiziertes Bit" können Sie ein spezifisches Bit innerhalb eines Byte-, Wort- oder Doppelwortsymbols ansprechen. Sie können ein globales Symbol nur mit einem globalen Bit und ein lokales Symbol nur mit einem lokalen Bit indizieren.

1. Öffnen Sie den Symbolmanager und klicken auf die Schaltfläche *Global hinzu* oder *Lokal hinzu*. Hierauf erscheint die Dialogbox *Symboldetails*.
2. Geben Sie den Namen des Bits im Feld *Symbolname* ein.
3. Wählen Sie den Datentyp BOOL im Feld *Typ*.
4. Das Kontrollkästchen *Indiziertes Bit* ist nur zugänglich, wenn Sie bereits mindestens einen der Symboltypen BYTE, WORD oder DWORD definiert haben. Aktivieren Sie das Kontrollkästchen *Indiziertes Bit*. Hierauf werden die Felder *Quelle* und *Bit #* aktiv.

5. Die Liste *Quelle* zeigt alle bereits definierten Symbole der Datentypen BYTE, WORD und DWORD (für globale oder lokale Symbole). Wählen Sie das Symbol, das Sie indizieren wollen, aus dem Feld *Quelle* aus.
6. Geben Sie die Bitnummer in das Feld *Bit #* ein. Mögliche Werte sind 0—7 für Bytes, 0—15 für Worte und 0—31 für Doppelworte.
7. Wahlweise können Sie im Feld *Beschreibung* eine Beschreibung eingeben.
8. Klicken Sie auf die Schaltfläche *OK*. Im Feld *Symbolliste* erscheint das indiziertes Bit.

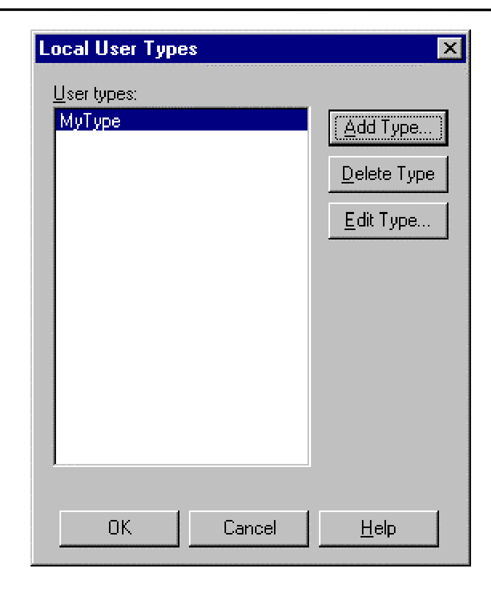
Anwenderdefinierte Datentypen editieren

Sie können einen Datentyp benutzerdefiniert anpassen. Ein anwenderdefinierter Datentyp besteht aus einer Gruppe von Datentypelementen (ganze Zahlen, reelle Zahlen, Zeichenfolgen usw.), die als eine Gruppe zusammenwirken. Die Datentypelemente brauchen nicht alle vom gleichen Datentyp zu sein. Sie definieren zum Beispiel einen Datentyp mit dem Namen Bericht. Er besteht aus drei Elementen mit den Namen Jahr, Aufgabe und Positionsnummern. Jahr und Positionsnummern sind ganze Zahlen, Auftrag ist eine Zeichenfolge. Sie können dann in jedem beliebigen Programm ein Symbol des Datentyps Bericht verwenden. Die darin enthaltenen Daten bestehen aus den drei Elementen, die ein Jahr, eine Aufgabenbezeichnung und eine Nummer darstellen.

Anwenderdefinierte Datentypen können global oder lokal sein. Ein lokaler Datentyp kann nur von dem Programm benutzt werden, das bei der Definition des Datentyps offen (aktiv) war.

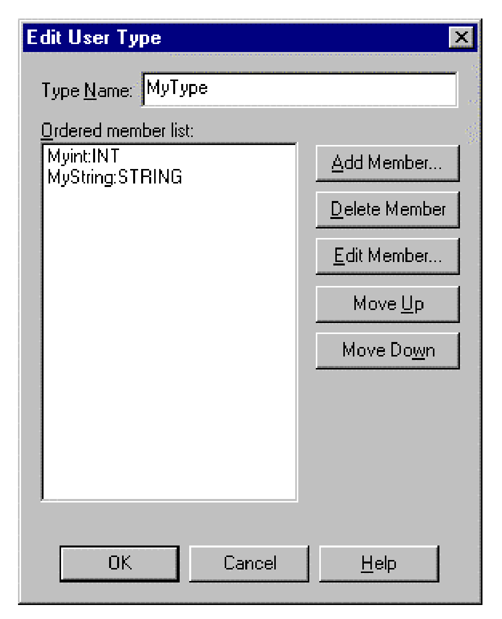
Editieren von anwenderdefinierten Datentypen:

1. Öffnen Sie den Symbolmanager.
2. Klicken Sie auf *Lokale Anwendertypen* oder *Globale Anwendertypen*. Hierauf erscheint die entsprechende Dialogbox *Anwendertypen*. Sie enthält die definierten Anwendertypen (lokal oder global) und enthält Schaltflächen, mit denen Anwendertypen hinzugefügt, editiert oder gelöscht werden können.



3. Führen Sie einen der folgenden Schritte durch:

   - Klicken Sie auf *Typ hinzu*, um einen neuen Typ zu erstellen. Hierauf erscheint die Dialogbox *Anwendertyp editieren*.
   - Zum Bearbeiten eines Anwendertyps wählen Sie die Typbezeichnung aus und klicken auf *Typ editieren*. Hierauf erscheint die Dialogbox *Anwendertyp editieren*.
   - Zum Löschen eines Anwendertyps wählen Sie die Typbezeichnung aus und klicken auf *Typ löschen*.



Feld	Beschreibung
Typenname	Zeigt den Anwendertyp an. Wollen Sie einen Anwendertyp hinzufügen, geben Sie den Typennamen ein.
Elementliste	Eine Liste der Typelemente.
Element hinzu	Fügt ein neues Element hinzu. Es erscheint eine Dialogbox *Typenelement editieren*, in der Sie dem Element einen Namen und einen Datentyp zuweisen können.
Element löschen	Löscht das ausgewählte Element.
Element editieren	Zum Editieren des ausgewählten Elements. Es erscheint eine Dialogbox *Typenelement editieren*, in der Sie dem Element einen Namen und einen Datentyp zuweisen können.
Aufwärts Abwärts	Bewegt die Elementauswahl in der Liste nach oben oder unten.

Symbole verwenden

Sie können bei Bedarf Symbolnamen von Hand über die Tastatur eingeben. Die Eingabe ist jedoch einfacher, wenn Sie die angezeigten Symbollistenboxen (zum Beispiel im Kontaktplaneditor) verwenden oder die Namen aus dem Symbolmanager ziehen (in Dokumente mit strukturiertem Text oder Anweisungslisten).

Drag&Drop

Mit den Drag&Drop-Funktionen des Symbolmanagers sind folgende Aktivitäten möglich:

- Sie können ein Boolesches Symbol auf einen Kontaktplan-Strompfad ziehen. Sie werden dann aufgefordert, das Symbol als einen Kontakt oder eine Spule einzufügen. Wenn Sie Kontakt oder Spule wählen, erscheint die Standard-Dialogbox *Kontakt editieren* bzw. *Spule editieren*.
- Sie können ein Boolesches Symbol auf einen im Kontaktplan bestehenden Kontakt oder eine Spule ziehen. Sie werden dann aufgefordert, das bestehende Kontakt- oder Spulensymbol mit dem vom Symbolmanager hergezogenen Symbol zu ersetzen.
- Sie können Symbole vom Symbolmanager direkt in eine Adresse des in strukturiertem Text oder als Anweisungsliste aufgebauten Dokuments ziehen.

Enumeration

Die Symbolelement-Auswahlliste erscheint bei Symbolen mit Gliederelementen (Strukturen, Funktionsblöcke oder Systemobjekte), wenn ein Symbol dieses Typs aus dem Symbolmanager in ein Dokument mit strukturiertem Text (zum Beispiel) gezogen wird. Sie können dann das Element, das Sie aus dieser Liste verwenden wollen, mittels Drag&Drop herausholen. Diese Vorgehensweise funktioniert auch, wenn Sie dem auf Seite 4-112 beschriebenen Überwachungsfenster Symbole hinzufügen (die Gliederelemente besitzen).

Systemsymbole

Vordefinierte Systemsymbole

Die Systemsoftware erzeugt automatisch die folgenden Symbole, die mit Anwenderprogrammen zusammen verwendet werden können.

Symbol	Beschreibung
TODAY	Enthält das aktuelle Systemdatum. Das Systemsymbol TODAY ist ein Datentyp DATE, der das aktuelle Systemdatum enthält. Mit ihm kann bestimmt werden, wann ein Ereignis stattfindet. Mit dem Symbol TODAY können folgende Operatoren verwendet werden: EQ, LT, GT, LE, GE und NE. Mit dem Zuweisungsbefehl oder dem Befehl MOVE können Sie für TODAY einen Wert festlegen. Mit dem Befehl ADD können Sie zu TODAY eine Zeitdauer addieren.
NOW	Enthält die aktuelle Systemzeit. Das Systemsymbol NOW ist ein Datentyp TOD, der die aktuelle Systemzeit enthält. Mit ihm kann bestimmt werden, wann ein Ereignis stattfindet. Mit dem Symbol NOW können folgende Operatoren verwendet werden: EQ, LT, GT, LE, GE und NE. Mit dem Zuweisungsbefehl oder dem Befehl MOVE können Sie für NOW einen Wert festlegen. Mit dem Befehl ADD können Sie zu NOW eine Zeitdauer addieren.
NULL	Mit diesem Symbol können Sie ein Pointersymbol auf einen Nullwert setzen oder ein Pointersymbol mit einem Nullwert vergleichen (gleich oder ungleich).
TMR-Variablen	Enthalten die Statusinformationen für den Datentyp TMR.
Zählervariablen	Diese Variablen enthalten Statusinformationen für Kontaktplanzähler (CTD, CTU und CTUD). Diese Symbole sind lokal zum Anwenderprogramm.
Timervariablen	Diese Variablen enthalten Statusinformationen für Kontaktplantimer (TOF, TON and TP). Diese Symbole sind lokal zum Anwenderprogramm.
"schrittname".X	Enthält den aktiv/inaktiv-Status eines Ablaufketten-Schritts. Diese Symbole sind lokal zum Anwenderprogramm.
"schrittname".T	Enthält die verstrichene Bearbeitungszeit eines Ablaufketten-Schritts. Diese Symbole sind lokal zum Anwenderprogramm.
Bewegungssteuerun g	Diese Variablen enthalten Statusinformationen für Achse, Achsenvariablengruppe, Programmsteuerung und Spindel.
Dateisteuerungsbloc k	Diese Variablen enthalten die Statusinformationen für den Dateioperationen.
Programmsteuerung	Enthalten die Statusinformationen für den Datentyp PRGCB.

Laufzeitsymbole

Die folgenden Symbole werden von der Systemsoftware automatisch erstellt. Sie sind über den Symbolmanager oder das Überwachungsfenster zugänglich und können in Anwenderprogrammen eingesetzt werden.

Symbolname	Beschreibung
RT_ERROR	(INT) arithmetische Fehler: 0 = fehlerfrei 1 = Division durch Null 2 = negative Quadratwurzel RT_ERROR muß vom Anwender gelöscht werden.
RT_FIRST_SCAN	(BOOL) wird vom ersten Zyklus des ersten Programms, das in der Laufzeitmaschine der PC-Steuerung abläuft, auf TRUE gesetzt. Sind alle Programme abgebrochen, wird RT_FIRST_SCAN wieder für den ersten Zyklus des ersten abzulaufenden Programms gesetzt.
RT_SCAN_OVERRUN	(BOOL) wird auf TRUE gesetzt, wenn E/A-Zyklus und Programmzyklus die Zykluszeit überschreiten.
RT_MAX_SCAN	(REAL) Dauer (in ms) des max. Laufzeitmaschinen-Zyklus.
RT_LAST_SCAN	(REAL) Dauer (in ms) des letzten Laufzeitmaschinen-Zyklus.
RT_AVG_SCAN	(REAL) Dauer (in Millisekunden) eines mittleren Zyklus der Laufzeitmaschine (Laufzeitzyklus= Programm + E/A + Overhead). Durchlaufender Mittelwert, über die letzten 100 Zyklen berechnet.
RT_LOGIC_MAX	(REAL) Dauer (in Millisekunden) des maximalen Programmzyklus.
RT_LOGIC_LAST	(REAL) Dauer (in Millisekunden) des letzten Programmzyklus.
RT_LOGIC_AVG	(REAL) Dauer (in Millisekunden) des mittleren Programmzyklus. Durchlaufender Mittelwert, über die letzten 100 Zyklen berechnet.
RT_IO_MAX	(REAL) Dauer (in Millisekunden) des maximalen E/A-Zyklus.
RT_IO_LAST	(REAL) Dauer (in Millisekunden) des letzten E/A-Zyklus.
RT_IO_AVG	(REAL) Dauer (in Millisekunden) des mittleren E/A-Zyklus. Durchlaufender Mittelwert, über die letzten 100 Zyklen berechnet.
RT_MEM_PCT	(REAL) enthält den verbleibenden Prozentwert des System-RAM (Heap-Speicher), der der SPS-Systemsoftware zugeordnet ist.
RT_LOW_BATTERY	(BOOL) Batterieausfallsignal von der USV.
RT_POWER_FAIL	(BOOL) Spannungsausfallsignal von der USV.
RT_SCAN_RATE	(REAL) konfigurierte Zykluszeit (in Millisekunden) entsprechend der Einstellung in der aktiven Konfiguration.

Schlüsselwörter

Die in der nachstehenden Tabelle aufgelisteten Bezeichner sind reservierte Systemsymbole (Schlüsselwörter). erstellen Sie mit diesen Bezeichnern keine Anwendersymbole. Beachten Sie, daß alle Systemsymbole in Großbuchstaben geschrieben sind. Diese Liste kann sich bei zukünftigen Ausgaben des Produkts ändern. Um Konflikte in der Zukunft zu vermeiden, sollten Sie für Anwendersymbole Kleinbuchstaben oder gemischt Groß- und Kleinbuchstaben verwenden.

Liste der Schlüsselwörter

ABORT_ALL	ABS	AC
ACCEL	ACOS	ACTION
ADD	ADD_NOFLY	AND_SLOWFLY
AND	AND_BITS	ANDN
ANDT	ANDTN	ANY
ANY_BIT	ANY_DATE	ANY_INT
ANY_NUM	ANY_REAL	APPENDFILE
ARRAY	ARRAY_TO_STRING	AS
ASIN	AT	ATAN
AXIS	AXISGRP	AXSJOG
BCD_TO_INT	BEGIN	BEGIN_IL
BEGIN_RS274	BOOL	BREAK
BY	BYTE	CASE
CAL	CALC	CALCN
CD	CLK	CLOSEFILE
CONCAT	CONFIGURATION	CONSTANT
COPYFILE	COS	CTD
CTU	CTUD	CU
CV	D	DATE
DATE_AND_TIME	DELETE	DELETEFILE
DINT	DIV	DO
DS	DSPMSG	DT
DWORD	ELSE	ELSEIF
EN	END	END_ACTION
END_CASE	END_CONFIGURATION	END_FOR
END_FOR_NOWAIT	END_FUNCTION	END_FUNCTION_BLOCK
END_IF	ENDIF	END_IL
END_RS274	ENO	END_PROGRAM

Liste der Schlüsselwörter

END_REPEAT	END_RESOURCE	END_STEP
END_STRUCT	END_TRANSITION	END_TYPE
END_VAR	END_WHILE	END_WHILE_NOWAIT
EQ	ESTOP	ET
EXIT	EXP	EXPT
F	FALSE	FB
F_EDGE	FILE	FIND
F_EDGE	FNAME	FOR
FROM	F_TRIG	FTYPE
FUNCTION	FUNCTION_BLOCK	G
GE	GLOBAL	GOTO
GT	H	I
IF	IN	IN1
IN2	INCLUDE	INIT
INITIAL_STEP	INSERT	INT
INTERVAL	INT_TO_BCD	INT_TO_REAL
INT_TO_STRING	IP	J
JMP	JMPC	JMPCN
JOGCONT	JOGDIR	JOGDIST
JOGHOME	JOGINCR	JOGMINUS
JOGPLUS	JOGSPD	JOGTYPE
K	L	LD
LDN	LDT	LE
LEFT	LEN	LIMIT
LINT	LL	LN
LOG	LREAL	LT
LWORD	M	MACROSTEP
MAX	MC	MID
MIN	MOD	MOVE
MOVEAXS	MS	MSGWND
MUL	MULP	MUX
N	NAME	NE
NEWFILE	NIL	NOT
NOW	NT	NULL
		OF
ON	OPENFILE	OR
OR_BITS	ORN	ORT
ORTN	OUT	P
PID	POSTN	POW

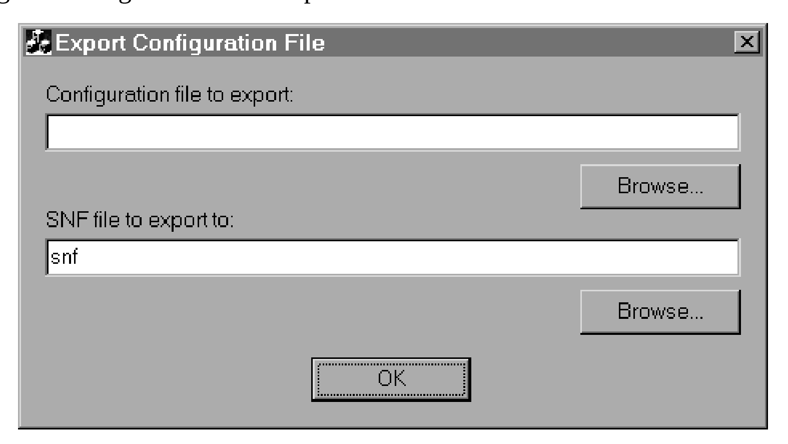
Liste der Schlüsselwörter

PRGCB	PRIORITY	PROGRAM
PT	PV	PW
Q	QU	QD
R	R1	READFILE
READ_ONLY	READ_WRITE	REAL
REAL_TO_STRING	R_EDGE	REPEAT
REPLACE	RESET_ESTOP	RESOURCE
RET	RETC	RETCN
RETAIN	RETURN	REWINDFILE
RIGHT	ROL	ROR
RS	RTC	R_TRIG
RUNG	S1	SCAN
SD	SEL	SEMA
SET_LOADSIZE	SHL	SHR
SIN	SINT	SL
SR	SQRT	ST
STT	STN	STTN
STEP	STOPJOG	STRING
STRING_TO_ARRAY	STRUCT	SUB
T	TAN	TASK
THEN	TIME	TIME_OF_DAY
TMR	TO	TOD
TODAY	TOF	TON
TP	TRANS	TRANSITION
TRUE	TRUNC	TYPE
UDINT	UL	ULINT
UNTIL	USINT	VAR
VAR_ACCESS	VAR_EXTERNAL	VAR_GLOBAL
VAR_INPUT	VAR_IN_OUT	VAR_OUTPUT
VEL	WHILE	WRITEFILE
WITH	WORD	XOR
XOR_BITS	XORN	XTON
XTOF	XTP	ZZZZ

Symbole für CIMPLICITY HMI exportieren

Symbole können im .snf-Format (Shared Name File) exportiert werden. Dieses Format ist bei GE Fanuc die SPS-spezifische Form von CSV.

Zum Export der Symboldatei gehen Sie in das Werkzeugmenü im Programmeditor und wählen Symbole für CIMPLICITY HMI exportieren. Hierauf erscheint die Dialogbox Konfigurationsdatei exportieren.



Wählen Sie über die Schaltflächen Durchsuchen die Konfigurationsdatei (.cfg) aus, aus der Sie exportieren wollen, sowie den Pfad für die .snf-Datei, zu der Sie exportieren wollen.

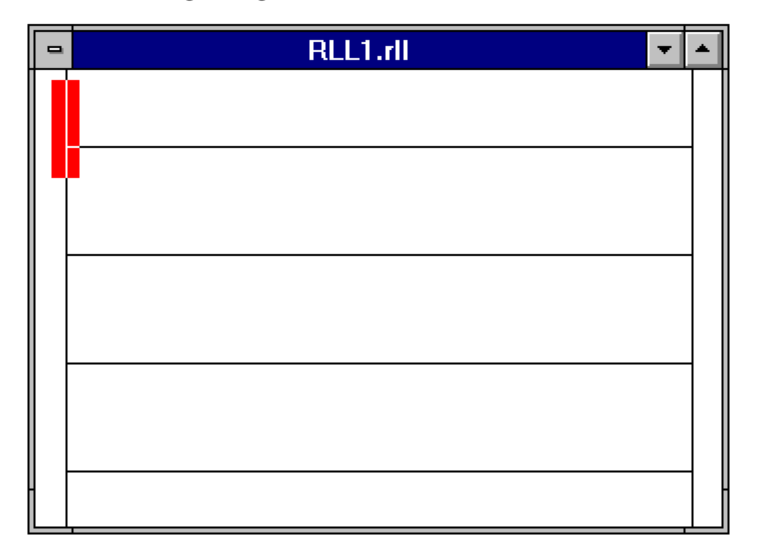
Einzelheiten über das Dateiformat .snf finden Sie in *CIMPLICITY HMI für Windows NT und Windows 95 Grundsystem, Anwenderhandbuch*, GFK-1180.

Abschnitt 2: Kontaktplanprogrammierung

Übersicht über Kontaktplanprogramme

Kontaktplanprogramme (KOP) werden üblicherweise in speicherprogrammierbaren Steuerungen zum Aufbau logischer Ablaufpläne verwendet. Kontaktplanprogramme sind so aufgebaut, daß sie den elektrischen Schaltplänen für einen äquivalenten elektrischen Relaisschaltkreis ähneln.

Das Kontaktplanprogramm enthält zwei vertikale Stromschienen. Die linke Stromschiene wird als die elektrische Stromquelle betrachtet und wird immer dann stromführend, wenn das Kontaktplanprogramm ausgeführt wird. Die Stromschiene auf der rechten Seite wird als die elektrische Stromsenke betrachtet. Die beiden Stromschienen sind durch horizontale Strompfade miteinander verbunden, in denen die logischen Anweisungen angeordnet sind.



Die elementaren Kontaktplanprogrammanweisungen, Kontakte und Spulen, stellen entweder echte Hardwarekomponenten dar (Endschalter, Spulen, Lampen, usw.) oder interne 1-Bit-Speicheradressen.

In den Kontaktplanprogrammen werden Eingangskontakte zur Darstellung Boolescher Eingangssymbole (Variablen) verwendet. Diese Kontakte wirken als

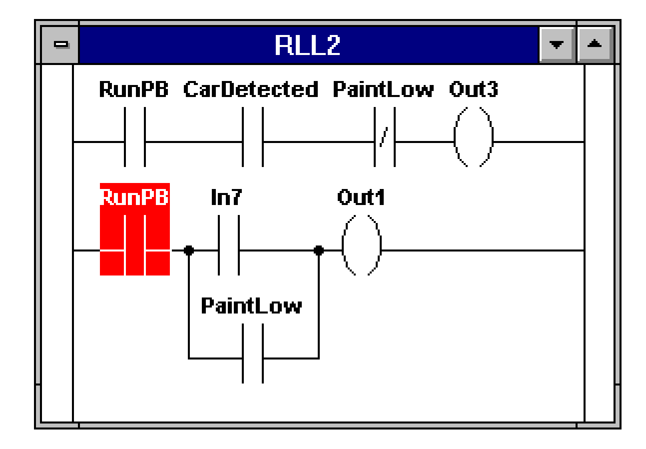
- Schließerkontakt (aktiv H-Pegel). Der Strom fließt zu dem Kontaktplanelement zur Rechten des Kontakts, wenn das mit dem Kontakt verknüpfte Symbol auf H-Pegel (1 oder EIN) ist.

- Öffnerkontakt (aktiv L-Pegel). Der Strom fließt zu dem Kontaktplanelement zur Rechten des Kontakts, wenn das mit dem Kontakt verknüpfte Symbol auf L-Pegel (0 oder AUS) ist.

Über den Zeichnungselementen dieser Eingänge stehen normalerweise die Namen der von ihnen repräsentierten Booleschen Symbole. Dies sind Variationen der Eingangskontakte, einschließlich Kontakte, die Umschaltvorgänge erkennen und während eines einzigen Zyklus des Kontaktplanprogramms Stromfluß zum nächsten Element weiterschalten.

In den Kontaktplanprogrammen werden Ausgangsspulen zur Darstellung Boolescher Ausgangsvariablen verwendet. Ist die Logik zur Linken einer Ausgangsspule durchgeschaltet (TRUE), empfängt das durch den Ausgang dargestellte Boolesches Symbol eine Boolesche 1 (H-Pegel); im anderen Fall empfängt es eine Boolesche 0 (L-Pegel). Es gibt Variationen des Spurenelements, einschließlich invertierender und pulsierter (Einzelzyklus) Ausgänge. Einige Ausgänge sind selbsthaltend und setzen (oder rücksetzen) die Ausgangsvariable nur dann auf TRUE (oder FALSE), wenn die Logik zu ihrer Linken durchgeschaltet (TRUE) ist.

Bei Kontaktplanelementen auf dem gleichen horizontalen Strompfad wird angenommen, daß sie über eine logisch UND-Funktion miteinander verbunden sind. Funktionsblöcke stehen für komplexere Funktionen zur Verfügung,die mehrere Boolesche Symbole enthalten.

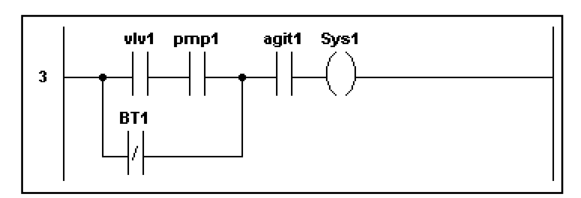


Lösen von Kontaktplan-Anwenderprogrammen

Hier wird erläutert, wie Kontaktplanprogramme mit einfachen Relaisfunktionen und Funktionsblöcken gelöst werden.

Wie einfache Relaislogik gelöst wird

Nachdem das System die physikalischen Ausgänge beschrieben hat, liest es die physikalischen Eingänge und löst dann die Kontaktplanlogik. Stromfluß und Lösung der Programmlogik erfolgt immer von oben nach unten und von links nach rechts. Im folgenden Beispiel beginnt der Stromfluß an der linken Stromschiene. Ist Kontakt vlv1 durchgeschaltet, geht der Stromfluß weiter zum Kontakt pmp1. Die drei Kontakte vlv1, pmp1 und agit1 liegen in Serie und stellen die logische UND-Verknüpfung der drei Kontakte dar. Kontakt bt1 liegt parallel zu den Kontakten vlv1 und pmp1. Dies stellt die logische ODER-Verknüpfung von bt1 mit vlv1 und pmp1 dar. Ist Kontakt BT1 durchgeschaltet, geht der Stromfluß weiter zu agit1, selbst wenn vlv1 nicht durchgeschaltet ist. Geht der Stromfluß zur Spule Sys1, wird der Kreis durchgeschaltet und der Stromfluß zur rechten Stromschiene kann stattfinden.

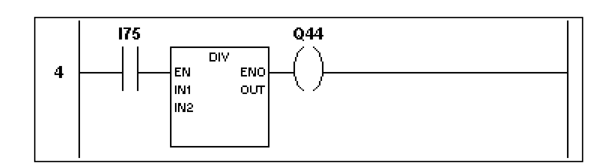


Wie Kontaktplanlogik bei Verwendung von Funktionsblöcken gelöst wird

Funktionsblöcke liefern einen Mechanismus zum Lösen komplexerer Operationen, die nicht auf einfache Weise mit Kontakten und Spulen gelöst werden können. Zum Beispiel:

- Arithmetische Operationen
- Logische Funktionen
- Zähler, Timer, usw.

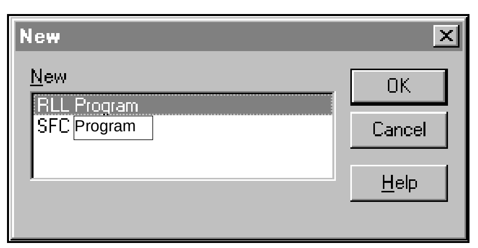
Ein Funktionsblock erhält über seine Eingänge Stromfluß vom Strompfad und gibt über seine Ausgänge Stromfluß an das nächste Element im Strompfad weiter. Ein Funktionsblock kann auch Daten aus seinen internen Eingängen lesen und zu den internen Ausgängen schreiben. Im folgenden Beispiel wird der Divisionsfunktionsblock (DIV) über seinen Strompfadeingang EN aktiviert. Dieser Eingang empfängt Stromfluß, wenn der Kontakt I75 aktiv ist. Der Funktionsblock erhält die Daten für Divisor und Dividend über zwei interne Eingänge (IN1, IN2). Der Funktionsblock schreibt den Quotienten zu einem internen Ausgang (OUT) und gibt über seinen Strompfadausgang (ENO) Stromfluß zum nächsten Programmelement Q44 weiter.



Ein Kontaktplanprogramm erstellen

Ehe Sie den Kontaktplaneditor öffnen und mit dem Erstellen eines Programms beginnen können, müssen Sie in die Programmentwicklungsumgebung gehen.

1. Starten Sie den Programmeditor.
2. Klicken Sie auf *Datei* und wählen dann *Neuer Editor*.Hierauf erscheint die Dialogbox *Neu*.



3. Wählen Sie *Kontaktplanprogramm* und klicken dann auf *OK*. Hierauf erscheint die Dialogbox *Speichern unter*.

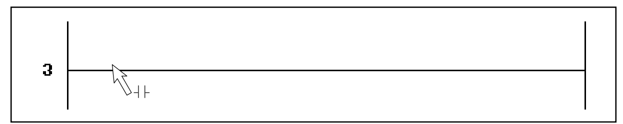


4. Bestätigen Sie den Standardwert oder geben Sie für das Programm einen neuen Namen ein und klicken dann auf *Speichern*. Die Standarderweiterung *.rll* wird angehängt,wenn die Datei in Ihrem Projekt gespeichert wird. Der Kontaktplaneditor zeigt eine neue Kontaktplandatei mit den beiden Stromschienen und einem Strompfad an.

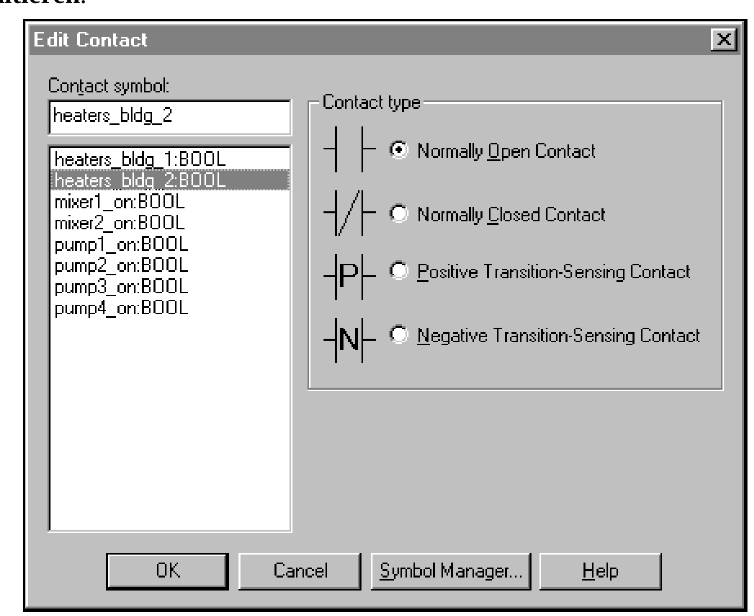
Programmelemente hinzufügen

Kontakt hinzufügen

1. Klicken Sie in der *Kontaktplan*-*Symbolleiste* auf das *Kontaktwerkzeug*. Der Cursor wird zum Kontaktcursor.
2. Setzen Sie den Cursor auf die Stelle im Strompfad, an der Sie den neuen Kontakt plazieren wollen.



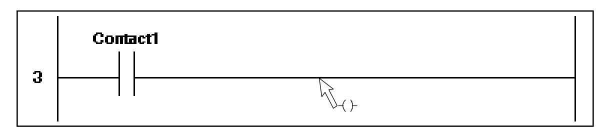
3. Klicken Sie die linke Maustaste. Hierauf erscheint die Dialogbox **Kontakt editieren**.



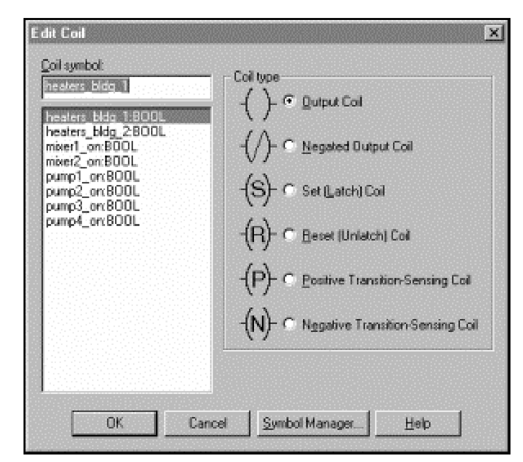
4. Haben Sie für Ihre System bereits die Symbolnamen definiert, dann klicken Sie jetzt auf das Symbol, das den Kontakt darstellen soll (heaters_bldg_2 in der Abbildung).
5. Haben Sie die Symbolnamen noch nicht definiert oder wollen Sie ein neues Symbol definieren, dann müssen Sie jetzt den Symbolmanager aufrufen und die Symboldaten für den Kontakt eintragen.
6. Wählen Sie den Kontakttyp (Schließer, Öffner, usw.) und klicken dann auf ***OK.*** Der neue Kontakt erscheint im Strompfad an der angegebenen Stelle.

Spule hinzufügen

1. Klicken Sie in der *Kontaktplan-Symbolleiste* auf das *Spulenwerkzeug*. Der Cursor wird zum Spulencursor.
2. Setzen Sie den Cursor auf die Stelle im Strompfad, an der Sie die neue Spule plazieren wollen.



3. Klicken Sie die linke Maustaste. Hierauf erscheint die Dialogbox **Spule editieren**.

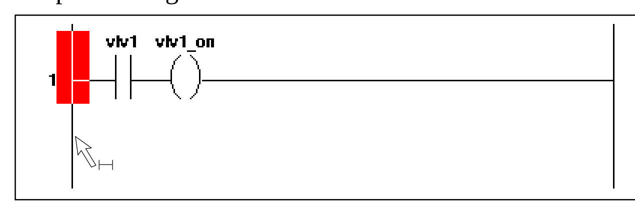


4. Haben Sie für Ihre System bereits die Symbolnamen definiert, dann klicken Sie jetzt auf das Symbol, das die Spule darstellen soll.
5. Haben Sie die Symbolnamen noch nicht definiert oder wollen Sie ein neues Symbol definieren, dann müssen Sie jetzt den Symbolmanager aufrufen und die Symboldaten für die Spule eintragen.
6. Wählen Sie den Spulentyp (Ausgang, negierter Ausgang, usw.) und klicken dann auf *OK***.** Die neue Spule erscheint im Strompfad an der angegebenen Stelle.

Sie können eine Ausgangsspule an jeder beliebigen Stelle im Strompfad setzen, auch zur Linken eines Eingangs oder in einem ODER-Zweig.**.** Eine Ausgangsspule speichert das logische Ergebnis der bis zu ihrem Platz im Strompfad ausgewerteten Logik.

Einen neuen Strompfad einfügen

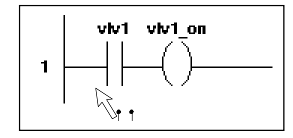
1. Klicken Sie auf das *Strompfadwerkzeug*. Am Mauszeiger hängt nun ein Strompfadsymbol.
2. Ziehen Sie den Cursor auf die Stelle der linken Stromschiene, an der Sie den neuen Strompfad einfügen wollen.



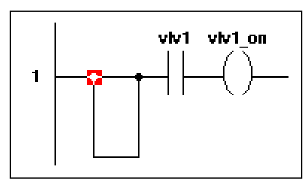
3. Klicken Sie die linke Maustaste. Der Editor fügt den Strompfad an der angegebenen Stelle ein.

Eine Verzweigung hinzufügen

1. Klicken Sie in der *Kontaktplan-Symbolleiste* auf das *Verzweigungswerkzeug***.** Der Cursor wird zum Verzweigungswerkzeug-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Strompfad, an der Sie die Verzweigung plazieren wollen.



3. Klicken Sie die linke Maustaste. Der Editor fügt die Verzweigung an der angegebenen Stelle ein.



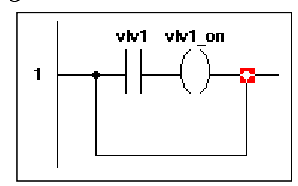
4. Nachdem die Verzweigung eingefügt wurde, können Sie die Kontaktpunkte nach Bedarf einstellen.

Kontaktpunkte einer Verzweigung verschieben:

1. Klicken Sie auf das *Auswahlwerkzeug*.

2. Setzen Sie den Cursor auf den Kontaktpunkt, den Sie verschieben wollen, und drücken Sie die linke Maustaste.
3. Halten Sie die Maustaste gedrückt und ziehen Sie Cursor und Kontakt auf die neue Stelle im Strompfad. Lassen Sie dann dort die Maustaste los.. Der Editor schließt die Verzweigung an der neuen Stelle am Strompfad an.

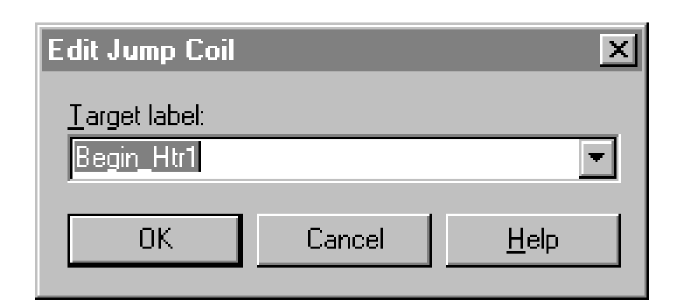


Sie können eine Verzweigung ohne Logik (einen Nebenschluß) einfügen. Mit einem Nebenschluß können Sie einen Logikbereich vorübergehend deaktivieren, ohne daß Sie ihn aus dem Programm löschen. In Zusammenhang mit einem Kontakt, der den Stromfluß zu der betreffenden Logik abschaltet, erhält der Nebenschluß den Stromfluß durch den Rest des Strompfads.. Diese Funktion ist hilfreich beim Austesten Ihres Programms.

Sie können einen Kontaktpunkt auch von einem Strompfad zu einem anderen verschieben, ohne die in der Verzweigung enthaltene Logik zu löschen.

Sprungmerker hinzufügen

1. Klicken Sie in der *Kontaktplan-Symbolleiste* auf das *Sprungwerkzeug***.** Der Cursor wird zum Sprungcursor.
2. Setzen Sie den Cursor auf die Stelle im Strompfad, an der Sie den neuen Sprung plazieren wollen.
3. Klicken Sie die linke Maustaste. Hierauf erscheint die Dialogbox **Sprungmerker editieren**.



4. Geben Sie eine Zielmarke ein und klicken dann auf *OK***.** Der Name einer Zielmarke darf keine Leerzeichen enthalten. **.** Der Editor fügt den Sprung an der angegebenen Stelle ein.

Einen Ablaufketten-Transitionsmerker einfügen

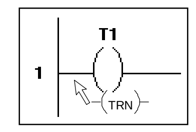
Der Ablaufketten-Transitionsmerker ist ein Kontaktplanprogrammelement, das Sie nur unter bestimmten Bedingungen (innerhalb einer Ablaufplanaktion) in einer Ablaufkette verwenden können**.**

Hinweis

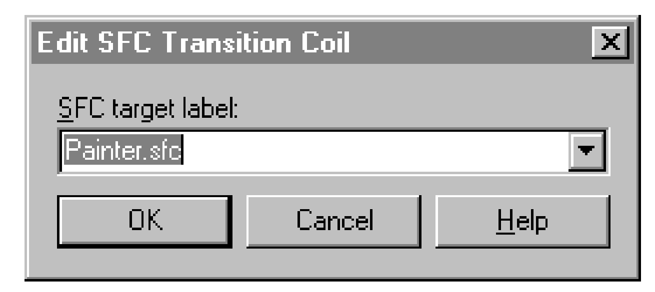
Um einen Ablaufketten-Transitionsmerker im Programm einfügen zu können, müssen Sie eine Aktion in einer Ablaufkette editieren**..**

Einfügen eines Ablaufketten-Transitionsmerkers:

1. Klicken Sie in der ***Kontaktplan-Symbolleiste*** auf den Ablaufketten-Transitionsmerker**.** Der Cursor wird zum Ablaufketten-Transitionsmerker-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Strompfad, an der Sie den Ablaufketten-Tansitionsmerker plazieren wollen.



3. Klicken Sie die linke Maustaste. Hierauf erscheint die Dialogbox **Ablaufketten-Transitionsmerker editieren**.



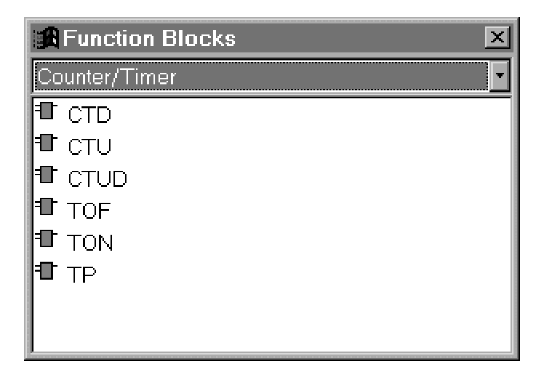
4. Geben Sie den Namen des Ablaufkettenziels ein und klicken dann auf ***OK.*** Der Name einer Ablaufkette darf keine Leerzeichen enthalten. **.** Der Editor fügt den Ablaufketten-Transitionsmerker an der angegebenen Stelle ein.

Funktionsblöcke hinzufügen

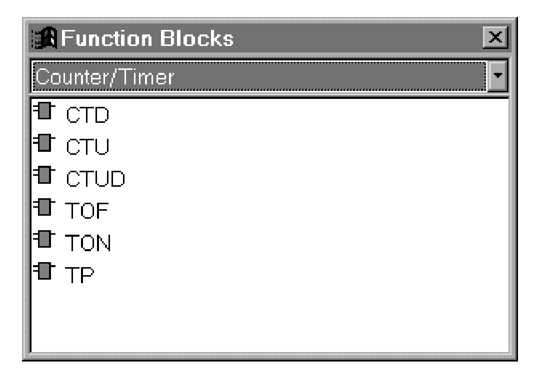
Es gibt einige vordefinierte Algorithmen, die Funktionsblöcke genannt werden, die Sie in einem Kontaktplanprogramm einsetzen können**.** Sie können einen Funktionsblock durch ein Eingangssignal von einem Kontaktplan-Strompfad aktivieren, ihn Operationen ausführen lassen (zum Beispiel trigonometrische, arithmetische oder logische Funktionen, Bitverschiebungen oder Dateioperationen) und die Ergebnisse dann über einen Ausgang an ein anderes Element im Kontaktplan-Strompfad weitergeben.

Einfügen eines Funktionsblocks:

1. Wird die Funktionsblockpalette nicht angezeigt, wählen Sie aus der Menüleiste *Anzeigen/Funktionsblockpalette*. Der Editor zeigt dann die Funktionsblockpalette an.

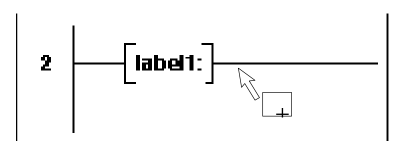


2. Wählen Sie den gewünschten Funktionsblocktyp aus der Drop-Down-Liste aus**.** Die Funktionsblockpalette verändert sich und zeigt die ausgewählten Funktionsblocktypen an.

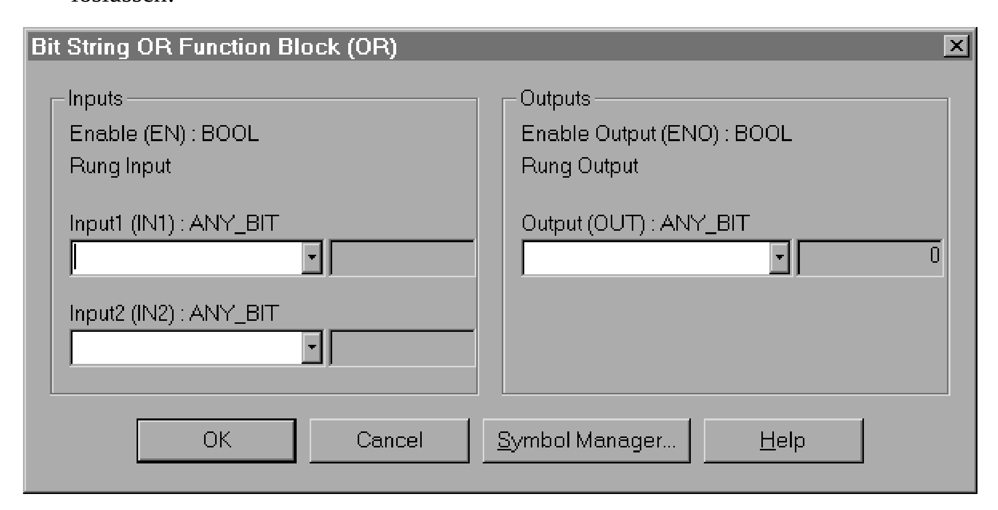


3. Klicken Sie auf den Funktionsblock, den Sie einfügen wollen (z.B. bitweise ODERn)**.**

4. Ziehen Sie den Funktionsblock an die Stelle im Strompfad, an der Sie ihn einfügen wollen.

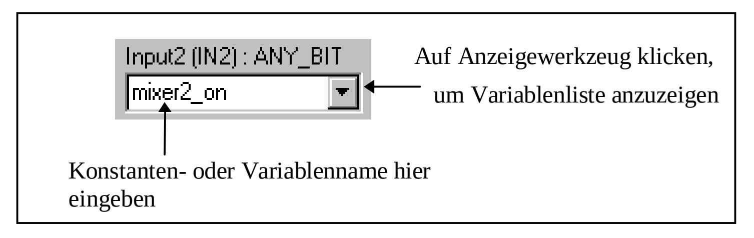


5. Die Dialogbox für den Funktionsblock erscheint, wenn Sie die linke Maustaste loslassen.

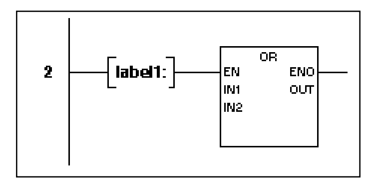


6. Tragen Sie die erforderlichen Daten für den Funktionsblock ein.

7. Wen Sie eine Konstante (ganze Zahl, reelle Zahl, Zeichen einer Zeichenfolge, usw.) eingeben, tippen Sie den Wert direkt in das Feld ein**.** Wenn Sie ein Symbol eingeben, tippen Sie entweder den Symbolnamen in das Feld ein oder klicken auf das Anzeigewerkzeug, um eine Liste der zulässigen Symbole zu Auswahl anzuzeigen.



8. Klicken Sie am Ende der Eingabe auf *OK*. Der Editor fügt den Block an der angegebenen Stelle ein.



Programmelemente verschieben und bearbeiten

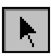
Programmelemente auswählen

Elemente über die Tastatur auswählen:

Wählen Sie das nächste Element in der gewünschten Richtung mit den Pfeiltasten aus. Bei einer unzulässigen Auswahl ertönt ein Piepton.

Elemente mit der Maus auswählen:

1. Wählen Sie in der *Kontaktplan-Symbolleiste* oder der *Ablaufplan-Symbolleiste* das *Auswahlwerkzeug* aus.

2. Setzen Sie den Cursor auf das gewünschte Element und drücken dann die linke Maustaste. Das Element wird in den eingestellten Hervorhebungsfarben hervorgehoben.

Spezielle Hinweise zur Auswahl einzelner Elemente

Um in einer Ablaufkette eine Auswahlverzweigung oder eine Simultanverzweigung auszuwählen, wählen Sie die obere oder untere Leiste der Verzweigung.

Um in einer Ablaufkette einen Strompfad auszuwählen, wählen Sie die linke oder rechte Stromschiene des Strompfads.

Mehrere Elemente gleichzeitig auswählen:

1. Wählen Sie in der *Kontaktplan-Symbolleiste* oder der *Ablaufplan-Symbolleiste* das *Auswahlwerkzeug* aus.



2. Setzen Sie den Cursor links über das am weitesten links stehende Element der gewünschten Gruppe (achten Sie darauf, daß er sich nicht auf einem Element befindet).
3. Drücken Sie die linke Maustaste und halten Sie sie gedrückt, während Sie den Cursor über die Elemente der gewünschten Gruppe ziehen. Durch das Ziehen des Cursors öffnet sich eine Auswahlbox.
4. Sind alle gewünschten Elemente in der Auswahlbox, lassen Sie die linke Maustaste los. Alle vollständig in der Auswahlbox enthaltenen Elemente sind nun selektiert.

Spezielle Hinweise zur Mehrfachauswahl

Um einen Verzweigungsanschluß in einem Kontaktplanprogramm auszuwählen, müssen beide Anschlüsse und alle Elemente der Verzweigung innerhalb der Auswahlbox liegen.

Um einen Strompfad in einem Kontaktplanprogramm auszuwählen, müssen die linke und die rechte Stromschiene sowie alle Elemente des Strompfads innerhalb der Auswahlbox liegen.

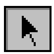
Um eine Schleife in einer Ablaufkette auszuwählen, müssen der oberste Schleifenpfeil, alle Schleifenübergänge sowie alle in der Schleife enthaltenen Elemente innerhalb der Auswahlbox liegen.

Um einen Sprung in einer Ablaufkette auszuwählen, müssen die Sprungraute, alle Übergänge sowie alle Zielmarken innerhalb der Auswahlbox liegen.

Um eine Auswahlverzweigung oder eine Simultanverzweigung in einer Ablaufkette auszuwählen, müssen die obere und untere Leiste sowie alle Elemente innerhalb der Auswahlbox liegen.

Eine Verzweigung verschieben

1. Wählen Sie in der *Kontaktplan-Symbolleiste* oder der *Ablaufplan-Symbolleiste* das *Auswahlwerkzeug* aus.



2. Setzen Sie den Cursor auf den gewünschten Verzweigungsanschluß. Drücken Sie die linke Maustaste und halten Sie sie gedrückt.
3. Ziehen Sie den Verzweigungsanschluß auf den gewünschten Punkt (der Cursor verändert seine Gestalt in einen Verzweigungsanschluß-Cursor).
4. Lassen Sie die linke Maustaste los, um den Verzweigungsanschluß am gewünschten Punkt abzulegen.

Drücken Sie die Taste <ESCAPE>, um den Ziehvorgang rückgängig zu machen.

Liegt der Zielpunkt des Verzweigungsanschlusses auf dem gleichen Strompfad und nicht innerhalb eines eingebetteten Zweiges oder außerhalb von verschachtelten Zweigen, dann wird der gezogene Verzweigungsanschluß am Ziehpunkt abgelegt und das andere Ende bleibt an seiner aktuellen Position. Liegt der Zielpunkt des Verzweigungsanschlusses jedoch auf einem anderen Strompfad, in einem eingebetteten Zweig oder außerhalb eines verschachtelten Zweigs, dann werden beide Enden der Verzweigung zu der Zielposition verschoben (es wird der gesamte Zweig verschoben).

Programmelemente verschieben

1. Wählen Sie die gewünschten Elemente aus und setzen Sie den Cursor auf eines dieser Elemente.
2. Drücken Sie die linke Maustaste und halten Sie sie gedrückt. Ziehen Sie dann die selektierten Elemente auf die gewünschte Stelle. Sobald das Ziehen beginnt, werden die gezogenen Elemente ausgeblendet und der Cursor stellt das Element dar, das gezogen wird. Werden mehrere Elemente gezogen, wechselt die Cursorgestalt in die des Gruppenziehcursors. Lassen Sie die linke Maustaste los, um die Elemente am gewünschten Punkt abzulegen.

Drücken Sie die Taste <ESCAPE>, um den Ziehvorgang rückgängig zu machen.

Programmelemente bearbeiten

1. Klicken Sie auf ***Auswahlwerkzeug***.
2. Doppelklicken Sie auf das Element (Schritt, Transition, usw.). Die dem Element (**Schritt bearbeiten**, **Kontaktplan-Übergangslogik auswählen, Sprungübergangslogik umgehen,** usw.) entsprechende Dialogbox erscheint.
3. Führen Sie die Änderungen in der Dialogbox entsprechend den Anforderungen durch.

Verzweigung löschen

Um ein Objekt mit dem Ausschneidewerkzeug zu löschen müssen Sie das Objekt selektieren und dann auf das Ausschneidewerkzeug klicken**.** Zum Löschen einer Verzweigung führen Sie folgende Schritte durch.

Löschen einer Verzweigung, die keine Elemente enthält:

1. Klicken Sie auf das *Auswahlwerkzeug*.
2. Setzen Sie den Cursor auf die Mitte der Verzweigung und klicken Sie.
3. Um die Verzweigung zu löschen klicken Sie auf das Ausschneidewerkzeug.

Löschen einer Verzweigung, die ein oder mehrere Elemente enthält

1. Klicken Sie auf das *Auswahlwerkzeug*.
2. Ziehen Sie eine Bereich, der den gesamten Zweig sowie dessen Anschlußpunkte enthält.
3. Um die Verzweigung zu löschen klicken Sie auf das Ausschneidewerkzeug.

Bearbeitungsvorgänge rückgängig machen/wiederherstellen

Um einen Vorgang rückgängig zu machen, klicken Sie in der *Symbolleiste* auf die Schaltfläche *Rückgängig*.

Kontaktplan-Anwenderprogramm dokumentieren

Ein Programmkommentar kann aus jeder Beschreibung bestehen, die Sie zu einem Strompfad anzeigen wollen. Sie können einstellen, ob das System die Kommentare anzeigt oder verbirgt.

Strompfadkommentare hinzufügen und bearbeiten

1. Klicken Sie in der Kontaktplan-Menüleiste auf das **Kommentarwerkzeug**.
   
   Ist einem Strompfad kein Kommentar zugeordnet, wird statt eines Kommentars **(* Strompfadkommentar *)** angezeigt.
2. Setzen Sie den Cursor auf den Strompfad im Programm, beidem Sie den Kommentar bearbeiten wollen, und doppelklicken Sie. Hierauf erscheint die Dialogbox **Programmkommentare**.
3. Geben Sie den Kommentar ein und klicken dann auf OK. Der Kommentar erscheint im angegebenen Strompfad.

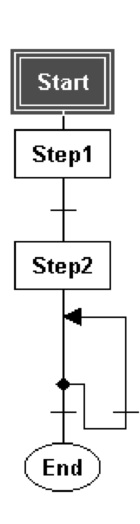
Symbolbeschreibungen hinzufügen und bearbeiten

1. Öffnen Sie ein Kontaktplan- oder Ablaufplanprogramm.
2. Klicken Sie auf das Symbolmanager-Werkzeug.
   
   Hierauf erscheint die Symbolmanager-Dialogbox für lokale/globale Symbole.
3. Fügen Sie Informationen hinzu oder bearbeiten Sie bestehende Daten.

Abschnitt 3: Ablaufketten-Programmierung

Übersicht über Ablaufpläne

Eine Ablaufkette stellt ein Anwenderprogramm als eine Reihe aufeinanderfolgender Schritte dar. Mit diesen Schritten, die Aktionen genannt werden, verknüpfen Sie Steuerlogik. Die Logik in den Aktionen wird ausgeführt, wenn der Schritt aktiv wird.



Schritte sind miteinander über Links verbunden; die Steuerung wird zwischen den Schritten durch Transitionen weitergegeben. Eine Transition kann ein Boolescher Ausdruck oder ein einzelner Kontaktplan-Strompfad sein.

Mit Verzweigungen können Sie mehrere Steuerpfade verwalten. Bei einer Auswahlverzweigung können Sie aus zwei oder mehr Steuerpfaden den einen auswählen, der aktiv sein soll. Bei Simultanverzweigungen können Sie mehrere Steuerpfade auswählen, die gleichzeitig gemeinsam aktiv sein sollen.

Es gibt noch weitere Programmlauffunktionen. Sie können Steuerschleifen einfügen, mit denen Sie eine Reihe von Schritten wiederholen können, oder über Sprung und Zielmarke die Kontrolle an eine andere Stelle übertragen können.

Sie können Ablaufpläne mit mehreren Pfaden erstellen. Mehrere Ablaufpläne können gleichzeitig aktiv sein.

Schritte

Ein Schritt stellt einen Zustand dar, in dem das Systemverhalten einem Regelsatz folgt, der von den mit dem Schritt verknüpften Aktionen und Funktionen definiert wird. Ein Schritt ist entweder aktiv oder inaktiv. Zu jeder Zeit wird der Systemzustand von dem Satz aktiver Schritte und den Werten seiner internen Variablen und Ausgangsvariablen bestimmt.

Innerhalb der Ablaufkette wird ein Schritt graphisch als ein Kästchen dargestellt. Dieses Kästchen enthält den Schrittnamen enthält, über den der Schritt identifiziert wird. Der Programmfluß in den Schritt hinein und wieder heraus erfolgt über eine senkrechte Linie, die oben in das Kästchen hineingeht und einer anderen Linie, die unten aus dem Kästchen herauskommt.

Wenn Sie eine neue Ablaufkette erstellen, generiert das System automatisch

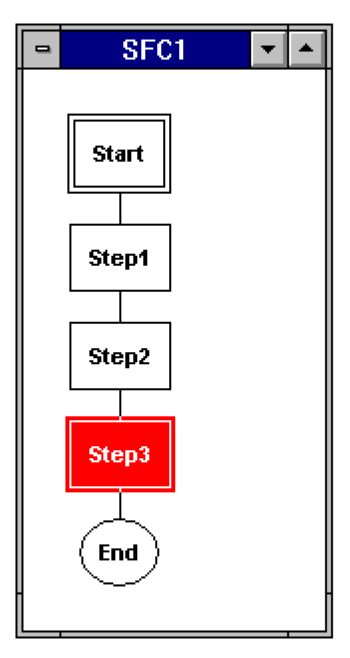
- den ersten Schritt mit der Bezeichnung Start,
- den letzten Schritt mit der Bezeichnung Ende.

Sie können diese Schritte nicht bearbeiten. Sie stellen lediglich Anfang und Ende der Ablaufkette dar.

Normalerweise trennen Sie Schritte in einer Ablaufkette durch Transitionen. Eine Transition ist ein Programmelement. In Erweiterung der Spezifikation IEC-1131-3, gestattet Ihnen die PC CONTROL Software, einen Schritt unmittelbar vor oder nach einem anderen Schritt zu plazieren. Zur Laufzeit fügt das System die erforderliche Dummy-Transition für Sie ein.

Ein Ablaufketten-Schritt kann mit einer oder mehreren Aktionen verknüpft sein. Eine Aktion kann einen oder mehrere Kontaktplan-Strompfade enthalten und über spezielle Aktions-Bestimmungszeichen die Ausführung einer Aktion steuern.

Aus einem Einzelschritt heraus können Sie eine weitere vollständige Ablaufkette (Tochter) zur Ausführung aufrufen— einen Makroschritt. Ist die Tochter-Ablaufkette fertig, geht die Programmsteuerung wieder zurück zu dem aufrufenden Makroschritt.



Schritteigenschaften

Um einen Schritt zu erzeugen, müssen Sie in der Schrittbearbeitungs-Dialogbox die Schritteigenschaften konfigurieren. Die nachstehende Tabelle faßt die Elemente zusammen, die konfiguriert werden können:

Feld/Schaltfläche:	Beschreibung:
Bewegungs-/Prozeßbefehle	Enthält den Programmcode des Schritts.
Strukturierter Text	Wählt strukturierten Text als Programmiersprache.
RS-274D	Wählt RS-274D als Programmiersprache. (Weitere Informationen siehe Anhang E.)
Datei verknüpfen	Verknüpft den Schritt mit einer Datei, die den Programmcode enthält.
Verknüpfte Datei editieren	Öffnet die verknüpfte Programmcode-Datei für den Schritt und zeigt sie in einem Editor an.
Symbolmanager	Ruft den Symbolmanager auf.
Anzeige (eine anklicken)	
Schrittname	Zeigt den Schrittnamen im Schritt an.
Bewegungs-/Prozeßbefehle	Zeigt den Code (strukturierter Text oder Bewegungssteuerung) im Schritt an.
Schrittbeschreibung	Zeigt die Schrittbezeichnung im Schritt an.
Symbol	Zeigt ein Symbol im Schritt an.
Breite	Stellt die Breite (in Pixel) für die Schrittbeschreibung ein.
Symbol...	Ruft die Symbolpalette auf.
Entfernen	Löscht ein zugewiesenes Symbol aus dem Schritt.

Verwendung der Schritt-Systemsymbole

Jeder Ablaufketten-Schritt besitzt zwei Systemsymbole (.X und .T). Sie können diese Systemsymbole in jedem Ausdruck, Kontakt oder Merker anstelle eines Symbol des gleichen Typs verwenden. Sie sprechen die Systemsymbole an, indem Sie den Schrittnamen eingeben, gefolgt von einem Punkt und dem Symbolsuffix.

Symbol	Definition	Beispiel
X	Boolescher Schritt-ist-aktiv Das Schritt-aktiv-Symbol X ist TRUE, wenn der Schritt aktiv ist. Es ist FALSE, wenn der Schritt inaktiv ist.	STEP1.X Bezieht sich auf das Schritt-aktiv-Symbol für den Schritt STEP1.
T	Schrittdauer Das Schrittdauersymbol T enthält die aktuell verstrichene Zeit des Schritts in ms. Bei inaktivem Schritt enthält T die insgesamt aufgelaufene Zeit des Schritts. T wird auf Null gesetzt, wenn der Schritt aktiv wird.	STEP1.T Bezieht sich auf das Schrittdauersymbol für den Schritt STEP1.

Über Aktionen

Aktionen enthalten Kontaktplanlogik. Aktionen werden graphisch als Rechtecke dargestellt, die den Namen der Aktion enthalten. Dieses Kästchen ist mit dem Schritt über eine waagerechte Linie verbunden. Sie können mehrere Aktionen mit einem Schritt verknüpfen.

Mit einem Schritt verknüpfte Aktionen werden aufgerufen, wenn der Schritt aktiv wird. Über ein Aktions-Bestimmungszeichen können Sie angeben, wann eine Aktion relativ zur Schrittaktivierung ausgeführt wird.

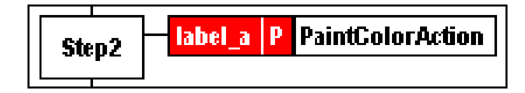
- Das Aktions-Bestimmungszeichen legt fest. wann der Kontaktplan relativ zur Aktivierung des Schritts, der in strukturiertem Text oder in Bewegungssteuerungssprache sein kann, durchlaufen wird. Sie können ein Aktions-Bestimmungszeichen mit oder ohne Bewegungs-Bestimmungszeichen verwenden. Eine Liste der Aktions-Bestimmungszeichen finden Sie auf Seite 4-49.
- Bewegungs-Bestimmungszeichen— legen fest, wann der Kontaktplan in Relation zur Ausführung des Bewegungssteuerungscode innerhalb des Schritts abläuft. Eine Liste der Bewegungs-Bestimmungszeichen finden Sie auf Seite 4-49.

Die Verwendung einer Programmmarke kann die Ausführung einer Aktion ebenfalls beeinflussen. Die Aktion wird erst ausgeführt, nachdem die Marke im Schrittcode aufgetreten ist.

Das Aktions-Bestimmungszeichen wird als ein Kästchen dargestellt, bei dem auf der rechten Seite der Aktion eine Abkürzung steht.

Aktionsfunktion

Im nachstehenden Beispiel besteht die Aktion mit dem Namen PaintColorAction aus mehreren Kontaktplan-Strompfaden. Diese Pfade werden ausgeführt, wenn Step2 aktiv wird. In diesem Beispiel beginnt die Ausführung des Kontaktplans erst, wenn die Codeausführung im Schritt auf die Marke label_a trifft. Der P Code ist das Aktions-Bestimmungszeichen, das aussagt, daß der Kontaktplan einmal ausgeführt wird (gepulst).



Mit einem Schritt können Sie keine, eine oder mehrere Aktionen verknüpfen. Über die Aktionsbezeichnung können Sie eine Aktion auch mit mehreren Schritten verknüpfen.

Der in einer mit einem Schritt oder einem Makroschritt verknüpften Aktion verwendete Ablaufketten-Transitionsmerker bewirkt folgenden Programmablauf:

- In Zusammenhang mit einem Schritt— Erhält der Ablaufketten-Transitionsmerker Stromfluß, dann wird der strukturierte Text in dem Schritt aufgehoben und die Programmausführung springt zu der im Ablaufketten-Transitionsmerker angegebenen Ablaufketten-Marke.
- In Zusammenhang mit einem Makroschritt— Erhält der Ablaufketten-Transitionsmerker Stromfluß, dann wird die von dem Makroschritt aufgerufene Tochter-Ablaufkette aufgehoben und die Programmausführung in der Mutter-Ablaufkette (der Ablaufkette mit dem Makroschritt) springt zu der im Ablaufketten-Transitionsmerker angegebenen Ablaufketten-Marke.

Aktionsparameter

Um eine Aktion zu erstellen, müssen Sie Aktionsparameter in der Dialogbox "Aktion bearbeiten" konfigurieren. In der nachstehenden Tabelle werden die Parameter zusammengefaßt.

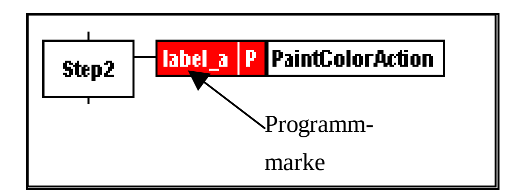
Feld/Schaltfläche	Beschreibung
Programmmarke	(Option) Der Kontaktplancode läuft erst ab, wenn der Code im Schritt auf die Marke trifft.
Aktions-Bestimmungszeichen	Gibt ein Aktions-Bestimmungszeichen an.
Bewegungs-Bestimmungszeichen	Gibt ein Bewegungs-Bestimmungszeichen an. Wählen Sie "Keine" für kein Bestimmungszeichen. Weitere Informationen finden Sie unter "Bewegungs-Bestimmungszeichen.”
Zeitdauer	(Nur Aktions-Bestimmungszeichen) gibt die Zeitdauer für begrenzte und Verzögerungs-Bestimmungszeichen an. Verwendet das Aktions-Bestimmungszeichen die Zeitdauer nicht, werden alle für diesen Parameter angegebenen Werte ignoriert.
Zeitdauer angeben	Ruft die Dialogbox "Zeitdauer definieren" auf.
Aktionsbezeichnung	Gibt den Namen der Aktion an.

Programmmarke

Wenn Sie eine (wahlweise) Programmmarke angeben, dann wird der Kontaktplancode in der Aktion erst ausgeführt, wenn der Schritt auf die Marke trifft. Aktion und Programmmarke müssen in der selben Ablaufkette stehen. Zwischen Programmen sind keine Querverweise erlaubt. Enthält der Schritt, mit dem die Aktion verknüpft ist, keine Marke, dann wird der Parameter "Programmmarke" ignoriert.

Die Marken in einer Makroschritt-Ablaufkette können von der Mutter-Ablaufkette aus nicht angesprochen werden. Ebenso können die Marken in der Mutter-Ablaufkette nicht von der Makro-Ablaufkette angesprochen werden.

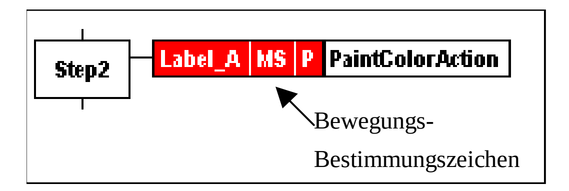
Bei einem Schritt mit strukturiertem Text besteht die Marke aus einem Markennamen, dem zwei Doppelpunkte folgen. Zum Beispiel: Marke::. Bei einem Schritt mit Bewegungssteuerungscode ist die Marke ein N, dem eine Satznummer folgt. Zum Beispiel: N45. Eine Marke erscheint in der Aktion wie folgt:



Bewegungs-Bestimmungszeichen

Hinweis Nicht alle Bewegungs-Bestimmungszeichen werden von allen Bewegungsmodulen unterstützt. In der Dokumentation zum Bewegungsmodul finden Sie Hinweise auf die unterstützten Bewegungs-Bestimmungszeichen.

Bewegungs-Bestimmungszeichen geben Bewegungsbedingungen an, die erfüllt sein müssen, damit der Kontaktplan ablaufen kann. Bei Verwendung einer Programmmarke, gilt die Bewegungsbedingung für den Bewegungssatz, der auf die Programmmarke folgt. Bestimmungszeichen erscheinen innerhalb die Aktion wie folgt:



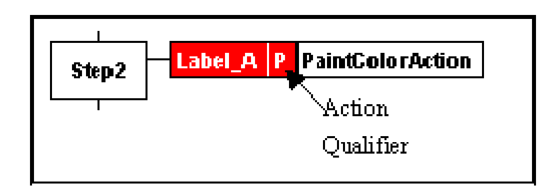
Wählen Sie eines der nachstehenden Bestimmungszeichen aus. Lassen Sie ein Leerzeichen, wenn Sie kein Bestimmungszeichen auswählen.

Soll der Kontaktplan erst bearbeitet werden, wenn ...	**... dann verwenden Sie dieses Bestimmungszeichen:**
die Bewegung beginnt	Bewegung gestartet (MS)
das Beschleunigungsprofil beendet ist	Beschleunigung beendet (AC)
die Bewegung die Sollgeschwindigkeit erreicht hat	Geschwindigkeit erreicht (AS)
die Bewegung mit dem Bremsprofil beginnt	Verzögerung gestartet (DS)
die Bewegung beendet ist	Bewegung beendet (MC)

die Bewegung beendet ist und alle mit der Bewegung verknüpften Achsen sich innerhalb der Lagetoleranz der programmierten Endpunktes befinden.	In Position (IP)
der Verfahrbefehl beendet ist.	Satzende (EB)

Aktions-Bestimmungszeichen

Aktions-Bestimmungszeichen geben Bedingungen für die Ausführung des Kontaktplancodes an. Bestimmungszeichen erscheinen innerhalb die Aktion wie folgt:



Wünschen Sie, daß nach aktiv werden des Schritts ...	**... dann verwenden Sie dieses Bestimmungszeichen:**
der Kontaktplan bearbeitet wird und die Bearbeitung stoppt, wenn der Schritt inaktiv wird.	nicht gespeichert (N)
der Kontaktplan bearbeitet wird, bis er durch das Rücksetz-Bestimmungszeichen rückgesetzt wird.	gespeichert (S)
der Kontaktplan einmal bearbeitet wird.	gepulst (P)
der Kontaktplan nach einer Verzögerung* bearbeitet wird. die Kontaktplanbearbeitung stoppt, wenn der Schritt inaktiv wird.	Zeitverzögerung (D)
die Kontaktplanbearbeitung läuft und stoppt, wenn das Zeitlimit* abläuft oder der Schritt inaktiv wird.	zeitbeschränkt (L)
die Kontaktplanbearbeitung nach einer Verzögerungszeit* begonnen wird und solange läuft, bis sie durch das Rücksetz-Bestimmungszeichen rückgesetzt wird. Setzt ein anderes Aktions-Bestimmungszeichen die Kontaktplanbearbeitung während der Zeitverzögerung zurück, hat dieses Rücksetzen keine Auswirkung, da der Kontaktplan noch nicht gespeichert war. Wird der Schritt inaktiv, ehe die Verzögerungszeit abgelaufen ist, wird der Kontaktplan nicht gespeichert und nie ausgeführt.	verzögert und gespeichert (DS)
die Kontaktplanbearbeitung nach einer Verzögerungszeit* begonnen wird und solange läuft, bis sie durch das Rücksetz-Bestimmungszeichen rückgesetzt wird. Wird eine Aktion während einer Verzögerung rückgesetzt, dann wird der Kontaktplan nicht bearbeitet.	gespeichert und zeitverzögert (SD)
die Kontaktplanbearbeitung begonnen wird und nach der angegebenen Zeit* stoppt. Zum Rücksetzen des Kontaktplans wird ein Rücksetz-Bestimmungszeichen benötigt. Ohne Rücksetzen würde der	gespeichert und zeitbeschränkt (SL)

Wünschen Sie, daß nach aktiv werden des Schritts ...	... dann verwenden Sie dieses Bestimmungszeiche n:
Kontaktplan nicht wieder bearbeitet. Wird der Schritt inaktiv, wird die Kontaktplanbearbeitung so lange fortgesetzt, bis die Zeit abgelaufen ist. Es wird zwar kein Rücksetz-Bestimmungszeichen benötigt, zum Anhalten der Kontaktplanbearbeitung kann aber einer eingesetzt werden.	
die Kontaktplanbearbeitung begonnen wird und nach Ablauf der angegebenen Zeit* stoppt. Wird der Schritt inaktiv, wird die Kontaktplanbearbeitung so lange fortgesetzt, bis die Zeit abgelaufen ist. Es wird zwar kein Rücksetz-Bestimmungszeichen benötigt, zum Anhalten der Kontaktplanbearbeitung kann aber einer eingesetzt werden.	Impulsbreite (PW)

Der vom Bestimmungszeichen "gespeichert" (S) gestartete Kontaktplan wird vom Rücksetz-Bestimmungszeichen (R) beendet. Sie können das Rücksetz-Bestimmungszeichen verwenden in einer anderen mit dem gleichen Schritt verknüpften Aktion oder in einer Aktion, die mit einem anderen Schritt verknüpft ist. Der Kontaktplan wird zwischen den Schritten bearbeiten. Ist die Aktion mit einem anderen Schritt verknüpft, dann muß die Aktion den gleichen Namen besitzen wie die Aktion, die rückgesetzt werden soll.

Zeitdauer

Sie können entweder die Zeitdauer direkt eingeben oder auf **Zeitdauer angeben** klicken und die Zeitabschnitte in der Dialogbox eintragen. Einzelheiten zur Eingabe der Zeitdauer finden Sie bei den nachstehenden Regeln und Beispielen.

- Befolgen Sie die IEC 1131-3 Spezifikation, wenn Sie die Zeitdauer direkt eingeben:
- T#, TIME#, t#, time#, gefolgt von der Zeitangabe in Tagen, Stunden, Minuten und Sekunden.

Zeit	Format	TIME	Format
14,7 Tage	T#14.7d	4 Sekunden	Time#4s
2 Minuten 5 Sekunden	T#2m5s	1 Tag 29 Minuten	t#1d29m
74 Minuten*	Time#74m	1 Stunden 5 Sekunden 44 Millisekunden	T#1h5s44ms

*Die IEC 1131-3 gestattet ein Überlaufen der höchstwertige Einheit in einer Zeitangabe.

Wollen Sie die Dialogbox benutzen, geben Sie die Zeit entsprechend in die einzelnen Felder ein.

Hinweis

Wenn Sie für eine Aktion eine Zeitdauer angeben und ein Aktions-Bestimmungszeichen wählen, der nicht zeitabhängig ist, dann wird die Zeitangabe ignoriert.

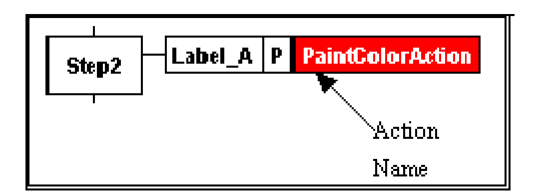
Wird bei einer Aktion sowohl eine Programmmarke als auch eine Zeitdauer angegeben, dann wird diese Zeit erst ab dem Zeitpunkt gezählt, an dem der Schrittcode an die Programmmarke kommt.

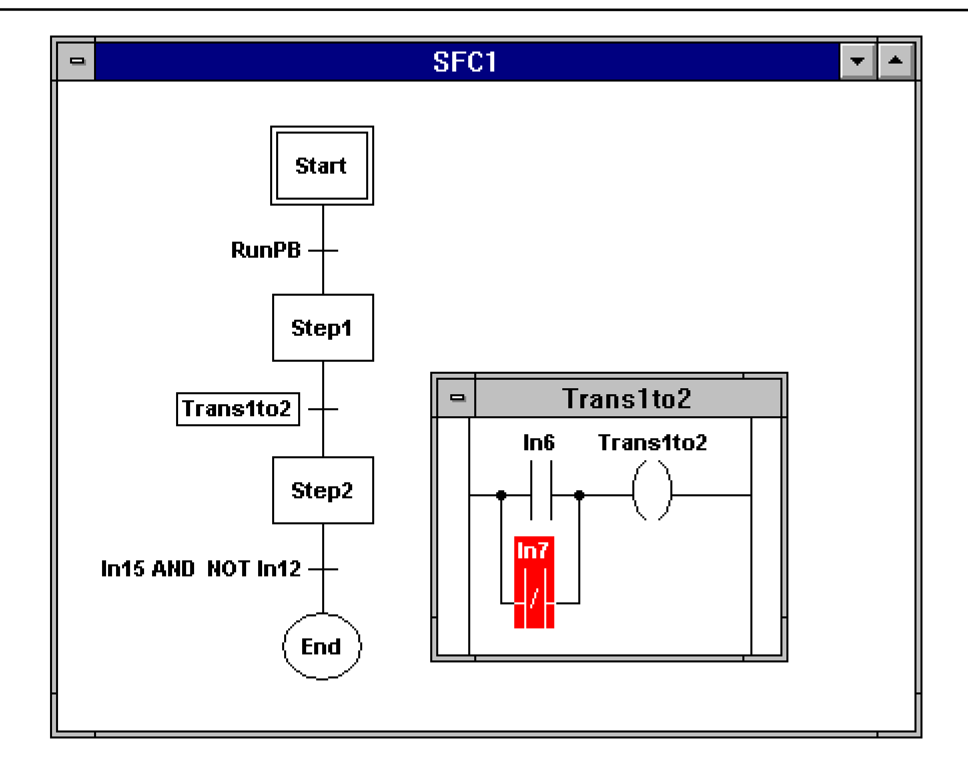
Aktionsbezeichnung

Mit der Aktionsbezeichnung wird der Name der Aktion gekennzeichnet. Verwenden Sie diesen Namen, wenn Sie auf die Aktion einer anderen Aktion aus zugreifen (wenn Sie zum Beispiel eine Aktion rücksetzen, die in einer anderen Aktion gespeichert wurde).

Eine Aktion kann den gleichen Namen wie eine Kontaktplantransition haben Die Kontaktplanlogik für Aktionen und für Kontaktplan-Transitionen hat unterschiedliche Bereiche. Die Verwendung gleicher Namen für Aktion und Transition bedeutet daher nicht, daß bei beiden die gleiche Kontaktplanlogik ausgeführt wird.

Die Aktionsbezeichnung erscheint in der Aktion wie nachstehend gezeigt:





Aktionsmanager

Mit dem Aktionsmanager (Auswahl "Aktionsmanager" im Werkzeugmenü) können Sie die mit Ablaufplan-Schritten verbundenen Aktionen verwalten. Der Aktionsmanager zeigt eine Liste aller in der aktiven Ablaufplandatei enthaltenen Aktionen an. In dieser Liste können Sie die angezeigten Aktionen umbenennen oder löschen.

Über Transitionen

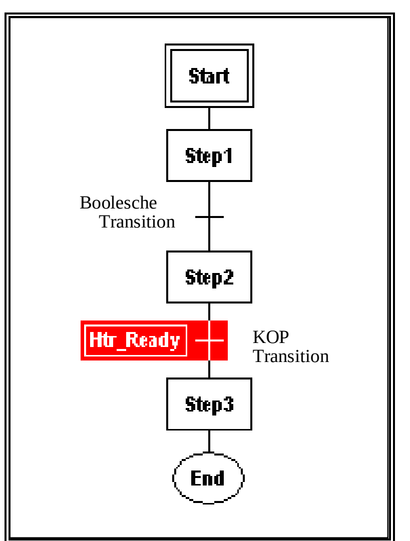
Eine Transition stellt den Zustand dar, bei dem der Programmlauf von einem oder mehreren Schritten, die vor der Transition stehen, zu einem oder mehreren Schritten nach der Transition entlang der entsprechenden gerichteten Verbindung fließt. Jede Transition besitzt eine mit ihr verknüpfte Transitionsbedingung, die das Ergebnis der Auswertung eines einzelnen Booleschen Ausdrucks darstellt. Wird der Code einer Transition ausgewertet, muß das Ergebnis entweder TRUE oder FALSE sein.

Mögliche Transitionsbedingungen sind:

- **Boolesche Transition -** Boolescher Ausdruck in strukturierter Textsprache. Wird dargestellt als horizontale Linie ohne Bezeichnung.

- **Kontaktplantransition -** benanntes Kontaktplanobjekt, das einen einzelnen Strompfad mit einer Ausgangsspule enthält, die den gleichen Namen wie das Transitionsobjekt trägt. Sie wird dargestellt als eine Linie (Kontaktplantransition) mit einer Markierung, die den Namen der Kontaktplan-Ausgangsspule enthält.

Der Schritt nach einer Transition kann nur ausgeführt werden, wenn die Transition vor ihm TRUE ist.



In dieser Abbildung hat der Programmfluß die Boolesche Transition und den darauf folgenden Step2 passiert und liegt momentan an der Kontaktplantransition an. Step3 kann erst ausgeführt werden, wenn die Kontaktplantransition TRUE ist.

Der Programmfluß in die Transition und aus ihr heraus erfolgt über eine Vertikalverbindung, die durch die Horizontalverbindung geht.

Die Norm IEC 1131-3 legt fest, daß in einem ein Ablaufplandiagramm zwischen jedem Schritt eine Transition und zwischen jeder Transition ein Schritt stehen muß. In Erweiterung der Norm gestattet Ihnen der Ablaufplaneditor, zwei Schritte oder zwei Transitionen unmittelbar aufeinander folgen zu lassen. Zur Laufzeit werden jedoch die erforderlichen "Dummyschritte" oder "Dummytransitionen" automatisch eingefügt.

Transitionsparameter

Transition	Parameter
Kontaktplan	Transitionsbezeichnung Kontaktplanlogik
Boolesche	Boolescher Ausdruck mit Operatoren und Symbolen

Über den Kontaktplantransitionsmanager

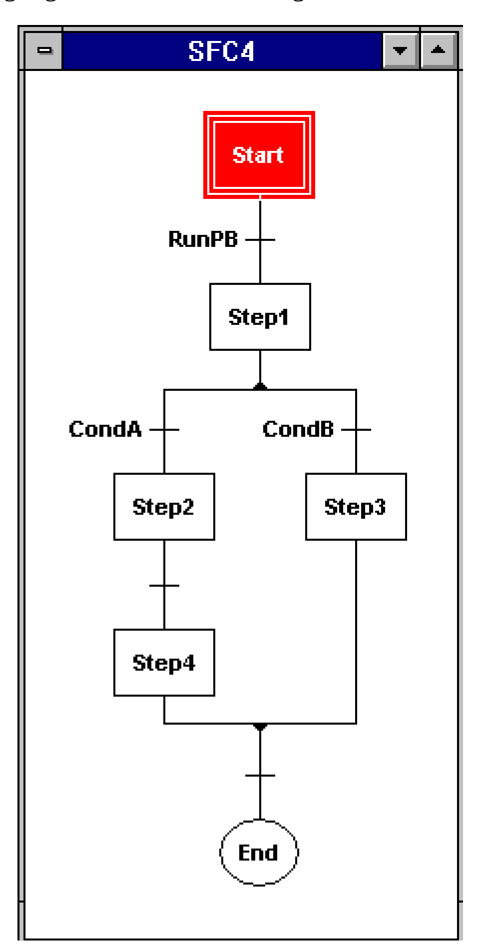
Mit dem Kontaktplantransitionsmanager können Sie die in einer Ablaufkette eingebetteten Kontaktplanaktionen verwalten. Der Kontaktplantransitionsmanager zeigt eine Liste aller in der aktiven Datei eingebetteten Kontaktplantransitionen an, die Sie in dieser Liste umbenennen oder löschen können.

Über Verzweigungen

Mit Verzweigungen können Sie in einer Ablaufkette mehrere Pfade steuern. Es gibt Auswahlverzweigungen und Simultanverzweigungen.

Auswahlverzweigungen

Über eine Auswahlverzweigung können Sie einen von mehreren Steuerpfaden auswählen. Jeder Pfad beginnt mit einer Transitionsbedingung die festlegt, welcher Pfad aktiviert wird. An einem bestimmten Punkt der Ablaufkette müssen alle Pfade der Auswahlverzweigung wieder in einen einzigen Pfad zusammenlaufen.

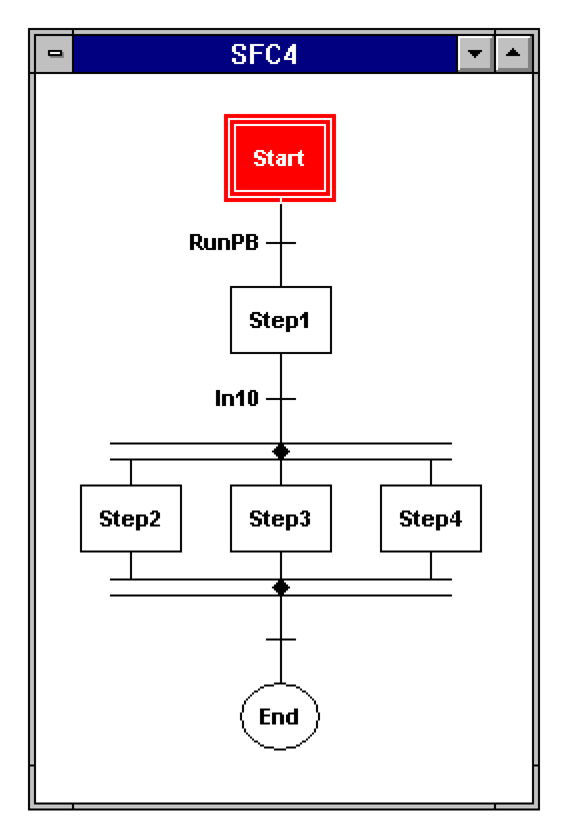


Simultanverzweigungen

Mit einer Simultanverzweigung können Sie gleichzeitig mehrere Steuerpfad bearbeiten. Alle Pfade der Simultanverzweigung werden aktiviert, sobald der Stromfluß die Verzweigung erreicht. An einem bestimmten Punkt der Ablaufkette müssen alle Pfade der Verzweigung wieder in einen einzigen Pfad zusammenlaufen. Die Zusammenführung muß jedoch warten, bis alle Pfade

- ausgeführt wurden, und
- am Zusammenführungspunkt angelangt sind.

Die Simultanverzweigung wird dargestellt als ein einzelner Pfad, der von oben hereinkommt, und einer doppelten horizontalen Linie mit zwei oder mehr nach unten austretenden Pfaden. Die Simultanzusammenführung wird dargestellt als zwei oder mehr Pfade, die von oben hereinkommen, einer doppelten horizontalen Linie und einem einzelnen nach unten austretenden Pfad.



Damit die Zusammenführung störungsfrei ablaufen kann, sollten Sie keine Marken verwenden, um

- aus einer Simultanverzweigung herauszuspringen,
- in eine Simultanverzweigung hineinzuspringen,

- zu einem anderen Pfad innerhalb einer Simultanverzweigung zu springen.

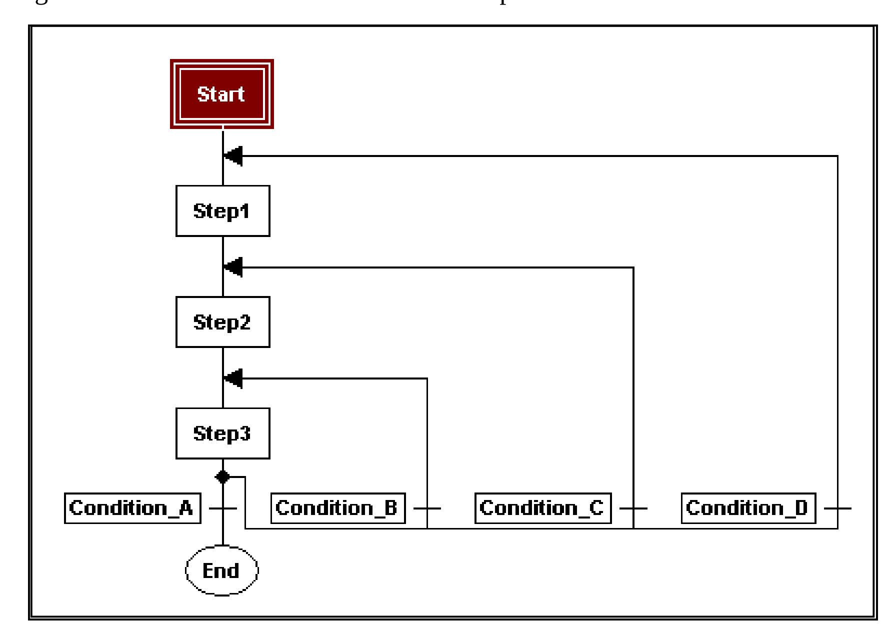
Programmfluß-Steuerungsfunktionen

Mit Steuerschleifen und Sprung- und Marken-Strukturen können Sie den Stromfluß in einer Ablaufkette beeinflussen.

Steuerschleifen

In einer Ablaufkette wird das Programm normalerweise von oben nach unten abgearbeitet. Mit Steuerschleifen können Sie zu einer bereits durchlaufenen Stelle zurückkehren und eine Reihe von Schritten wiederholen. Eine Steuerschleife besteht aus zwei Transitionen: Ein, die in Abwärtsrichtung weiterführt, und eine, in einer gerichteten Verknüpfung, die zurück nach oben führt. Ein Pfeil oben an der Steuerschleife zeigt an, wo der Stromfluß wieder in den Steuerpfad zurückkehrt.

In nachstehender Abbildung läuft der Stromfluß bis zum Ende, wenn Condition_A TRUE ist. Ist Condition_B TRUE und Condition_A FALSE, kehrt der Programmfluß zu dem Punkt oberhalb von Step3 zurück.



Bei Mehrfachverzweigungen wird die logische Auswertung von links nach rechts durchgeführt. Sind alle Transitionen FALSE, hält der Programmfluß solange an, bis eine der Transitionen TRUE wird.

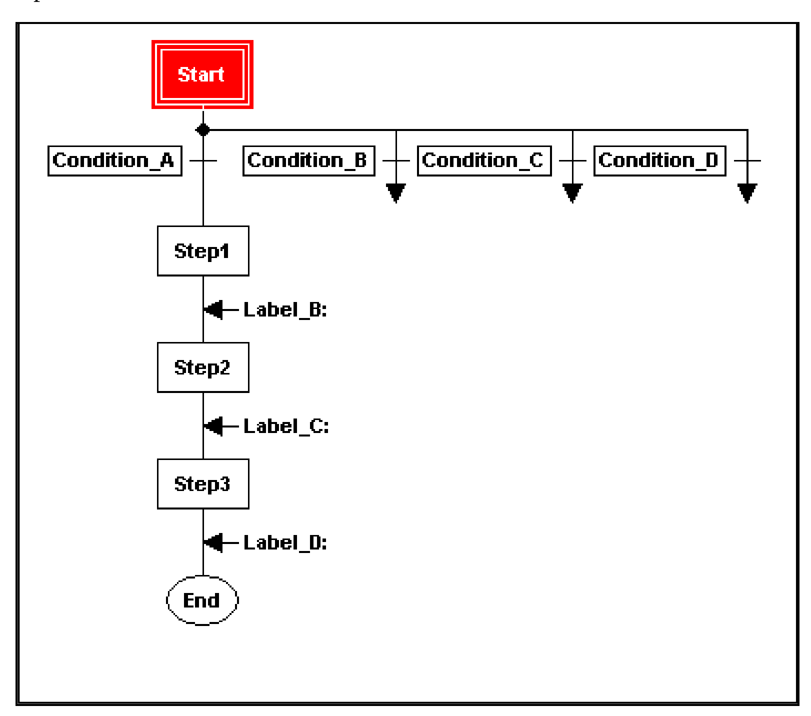
Mit einer der folgenden Methoden können Sie die Transitionen für die Schleife definieren:

- Boolesche Transition aus einem Boolescher Ausdruck aus strukturiertem Text.
- Kontaktplantransition aus einem einzelnen Kontaktplan-Strompfad mit einer Ausgangsspule, die den gleichen Namen wie die Transition selbst besitzt.

Sprung und Marke

Mit einer Sprung- und Markenstruktur können Sie den Stromfluß aus jedem beliebigen Teil einer Ablaufkette übertragen. Das Sprungelement wird durch zwei Transitionen dargestellt: Eine, die nach unten weiterführt, und eine zur Übertragung zu einem Markenbezeichner. Das Markenelement wird dargestellt durch einen Pfeil nach links, der den Punkt anzeigt, an dem der Stromfluß wieder in den Steuerpfad eintritt, und einem Markenbezeichner, dem ein Doppelpunkt folgt.

In nachstehender Abbildung läuft der Stromfluß zu Step1, wenn Condition_A TRUE ist. Ist Condition_B TRUE und Condition_A FALSE, springt der Programmfluß an Step1 vorbei zu Label_B. Ist Condition_C TRUE und Condition_A und Condition_B sind FALSE, springt der Programmfluß an Step1 und Step2 vorbei zu Label_C.



Bei Mehrfachverzweigungen wird die logische Auswertung von links nach rechts durchgeführt. Sind alle Transitionen FALSE, hält der Programmfluß solange an, bis eine der Transitionen TRUE wird.

Mit einer der folgenden Methoden können Sie die Transitionen für den Sprung definieren:

- Boolesche Transition aus einem Boolescher Ausdruck aus strukturiertem Text.
- Kontaktplantransition aus einem einzelnen Kontaktplan-Strompfad mit einer Ausgangsspule, die den gleichen Namen wie die Transition selbst besitzt.

Sprung- und Markenparameter

Feld	Beschreibung
Sprungzielmarke	Gibt die Marke an, zu der der Programmfluß weitergeschaltet wird.
Markenname	Gibt den Punkt in der Ablaufkette an, an dem der Stromfluß wieder aufgenommen wird.

Der Name einer Marke muß mit einem Buchstaben beginnen, dem alphanumerische Zeichen oder ein Unterstrich folgen.

Auswertung von Ablaufplänen

Der Programmfluß in einer Ablaufkette bewegt sich von oben nach unten. Der Code in den einzelnen Schritten wird bearbeitet. Ist der Code abgearbeitet, geht der Programmfluß weiter zum nächsten Programmelement.

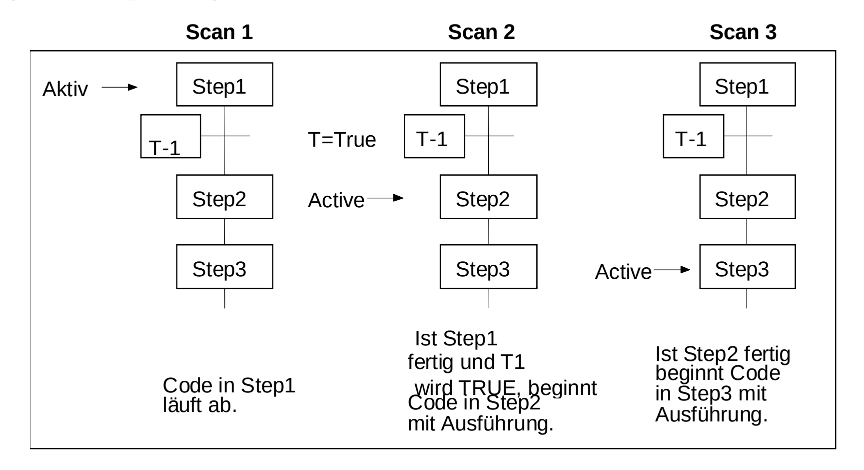
Ist das nächste Element ...	... dann ...
ein Schritt	wird der Code in diesem Schritten bearbeitet.
eine Transition	wird der Programmfluß weitergeschaltet, wenn die Transition TRUE wird.

Jeder Schritt in einer Ablaufkette benötigt bis zum Abschluß mindestens zwei E/A-Zyklen. Im ersten Zyklus werden alle mit dem Schritt verknüpften Aktionen ausgewertet. Liegen keine Schleifenstrukturen (z.B. FOR, WHILE, REPEAT) vor, dann werden alle Programmteile aus strukturiertem Text fertig bearbeitet. Die Transition wird geprüft, nachdem die gesamte Logik ausgewertet wurde (im zweiten Zyklus).

Enthält ein Programm in strukturiertem Text Schleifenelemente, dann wird das Programm bis zum Ende ausgewertet und es erfolgt ein Sprung zurück zum Beginn des nächsten E/A-Zyklus. Eine Transition wird erst geprüft, nachdem der strukturierte Text vollständig bearbeitet wurde. Mit den strukturierten Textelementen END_FOR_NOWAIT und END_WHILE_NOWAIT können Sie die Schleifenbearbeitung in einem E/A-Zyklus beenden.

Selbst wenn es der Ablaufplaneditor zuläßt, daß Sie die einzelnen Schritte unmittelbar hintereinander plazieren, benötigt jeder Schritt zur vollständigen Bearbeitung immer noch **mindestens zwei E/A-Zyklen**, da bei der automatischen Syntaxanalyse zwischen den Schritten "Dummytransitionen" eingefügt werden. Diese Transitionen werden immer als TRUE ausgewertet. Ihre Anwesenheit wird durch die Norm IEC 1131-3 gefordert. Der Ablaufplaneditor hilft Ihnen, den Programmieraufwand zu reduzieren, indem er diese Transitionen automatisch in das syntaxgeprüfte Programm einfügt.

Im folgenden Beispiel folgt ein Schritt dem anderen. Der Programmfluß geht immer noch von oben nach unten und die Bearbeitung eines Programmelements beginnt erst, nachdem das vorherige Element fertig bearbeitet wurde. Step2 wird aktiv nachdem die erste Transition TRUE wurde. Step3 wird aktiv,nachdem die Logik von Step2 fertig bearbeitet wurde.



Die Software gestattet die Bearbeitung mehrere Programme unterschiedlichen Typs. Sie können zum Beispiel zur gleichen Zeit zwei Kontaktplanprogramme und drei Ablaufplanprogramme laufen lassen. Sie können die Programmausführung über über globale Variablen, die von allen Programmtypen erkannt werden, koordinieren.

Auswertung von Transitionen

Die gesamte Logik aus strukturiertem Text oder RS-274D in einem Schritt muß fertig bearbeitet sein, ehe das System eine Transition auswertet. Dies ist eine zusätzliche Erweiterung der IEC-1131-3 Spezifikation, bei der lediglich alle vorhergehenden Schritte aktiv sein müssen, damit eine Transition ausgewertet werden kann.

Eine Transition kann ohne zwischengeschaltete Schritte direkt auf eine andere Transition folgen. Ist eine Transitionsbedingung erfüllt, wird die Transition gesperrt und das nächste Ablaufkettenelement (Schritt oder Transition) wird aktiviert. Bei Laufzeit wird zwischen zwei aufeinanderfolgenden Transitionen ein

"Dummyschritt" eingefügt (entsprechend der IEC-Norm). Dieser Dummyschritt führt keine logische Funktion aus, benötigt aber für Aktivierung und Deaktivierung wie jeder normale Schritt auch zwei E/A-Zyklen.

Eine Transition wird wie folgt ausgewertet:

Diese Transition:	... wird TRUE wenn ...
Kontaktplantransition	Der Stromfluß auf dem Strompfad erreicht die Ausgangsspule und schaltet sie durch.
Boolesche Transitionen	Der Boolesche Ausdruck wird TRUE.

Ehe der Programmfluß zum nächsten Schritt fortgesetzt werden kann, müssen die Transitionen als TRUE ausgewertet werden.

Ist die Transition ...	... dann ...
TRUE	Beim nächsten E/A-Zyklus wird der mit der Aktion verknüpfte Kontaktplan bei abgeschalteter Spannung ausgewertet, um Ausgänge abzuschalten., Bei darauffolgenden E/A-Zyklus aktiviert der Stromfluß den nächsten Schritt.
FALSE	Der Schritt bleibt aktiv und vorhandene Kontaktplanlogik wird ausgewertet. Solange der Schritt aktiv ist, wird vorhandene Kontaktplanlogik ausgewertet. Logik im strukturierten Text wird jedoch nur einmal beim ersten E/A-Zyklus ausgewertet.

Ist eine Transition FALSE und bleibt FALSE, führt das System den strukturierten Text oder Bewegungscode in den Schritten vor der Transition nicht nochmals aus. Kontaktplanlogik wird jedoch erneut ausgeführt. Der Programmfluß liegt solange an der Transition an, bis sie TRUE wird.

Simultanverzweigungen

Beim Erstellen von Simultanverzweigungen:

- Sprechen Sie das gleiche Symbol nicht in unterschiedlichen Pfaden der Verzweigung als Ausgang an.
- Rufen Sie die gleiche Tochter-Ablaufkette nicht aus Makroschritten in unterschiedlichen Pfaden der Verzweigung auf.

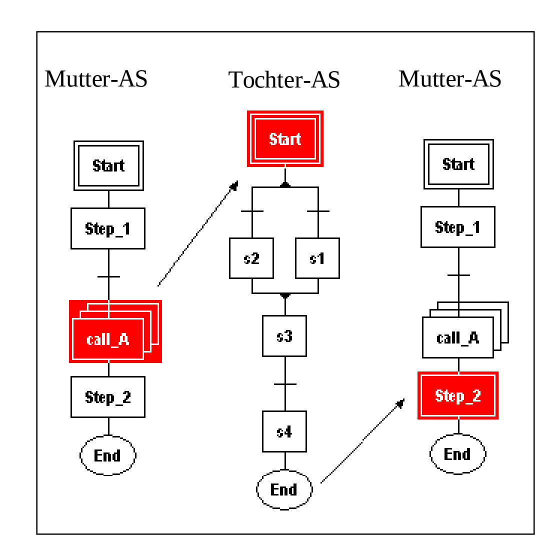
Damit die Zusammenführung störungsfrei ablaufen kann, sollten Sie keine Marken verwenden, um

- aus einer Simultanverzweigung herauszuspringen,
- in eine Simultanverzweigung hineinzuspringen,
- zu einem anderen Pfad innerhalb einer Simultanverzweigung zu springen.

Verwendung von Makroschritten

Makroschritte rufen ein Ablaufkette zur Ausführung von einem Schritt in einer anderen Ablaufkette heraus auf. Der Programmfluß geht zu der gerufenen Ablaufkette (Tochter-Ablaufkette). Ist die Bearbeitung der Tochter-Ablaufkette abgeschlossen, geht der Programmfluß zurück zur Mutter-Ablaufkette und wird nach dem Makroschritt fortgesetzt.

In der folgenden Abbildung ist call_A der Makroschritt in der Mutter-Ablaufkette, der die Tochter-Ablaufkette zur Ausführung aufruft. Ist die Bearbeitung der Tochter-Ablaufkette abgeschlossen, wird der Programmfluß bei Step_2 in der Mutter-Ablaufkette fortgesetzt.



Die Darstellung eines Makroschrittes erfolgt ähnlich wie bei einem Schritts mit mehreren Kästchen, die einen Bezeichner enthalten. Der Programmfluß in den Makroschritt hinein und wieder heraus erfolgt über eine senkrechte Linie, die oben hineingeht und einer anderen Linie, die unten herauskommt.

Mit einem weiteren Programmelement, der Aktion, können Sie die Codebearbeitung innerhalb des Makroschrittes mit anderen Programmcodeelementen koordinieren.

Wird in einer Mutter-Ablaufkette ein lokales Symbol definiert, dann kann die Tochter-Ablaufkette dieses lokale Symbol sehen. Im Symbolmanager für die Tochter-Ablaufkette können Sie dieses lokale Symbol jedoch nicht sehen.

Erweiterungen von IEC 1131-3

Im Vergleich zu den IEC-1131-3 Standard-Kontaktplanprogrammen und - Ablaufplänen enthält PC Control einige wahlweise einsetzbare Erweiterungen.

Erweiterungen von KOP

- Ausgangsspulen können an beliebiger Stelle eines Strompfads eingefügt werden,
  - Sie speichern das Boolesche Teilergebnis, das bis zu diesem Punkt ausgewertet wurde.
  - Sie geben den Stromfluß weiter zum nächsten Element zu ihrer Rechten.
- "OR" Zweige ohne logische Elemente sind erlaubt.
  - Sie können als "Nebenschluß" um einen Logikblock eingesetzt werden, ohne daß dieser dabei gelöscht wird.
  - Sie erzeugen Compilerwarnungen, die möglicherweise vergessene Elemente anzeigen.

Erweiterungen von AS

- Schrittkästchen können Befehle in RS-274D oder strukturiertem Text enthalten. Diese Befehle in einem Schrittkästchen werden nacheinander ausgeführt. Die Ausführung eines Schrittes ist erst abgeschlossen, wenn alle Befehle innerhalb eines Schrittes abgearbeitet sind. Die auf einen Schritt folgenden Transitionsbedingungen werden erst ausgewertet, nachdem der Inhalt des Schritts bearbeitet wurde.
- Schritte können entweder durch ein Kästchen dargestellt werden, das den Namen des Schrittbezeichners enthält, oder durch ein Kästchen, das die Befehlsliste des Schrittes enthält, oder durch ein Symbol, das aus einem Bitmapbild und einer oder zwei Zeilen Text besteht..
- Eine Symbolpalette enthält eine Sammlung vordefnierter Schritte, die aus einem Symbol und einer Liste von Befehlen innerhalb des Schritts besteht. Diese vordefinierten Schritte können für Bibliotheks- oder Festroutinen verwendet werden.
- Dateien können zum Einfügen in die Befehlsliste innerhalb eines Schrittkästchens markiert werden. Es können mehrere Dateien eingefügt werden. Programmbefehle können mit den Include-Befehlen gemischt werden.
- Ein Schritt kann ohne zwischengeschaltete Transition direkt auf einen anderen Schritt folgen. Folgt auf einen Schritt keine Transition, dann wird er

deaktiviert, sobald sein Inhalt ausgeführt wurde, und der nächste Schritt wird aktiviert. Bei Laufzeit wird zwischen zwei aufeinanderfolgenden Schritten eine "Dummytransition" eingefügt (entsprechend der IEC-Norm). Die Dummytransition wird immer mit TRUE ausgewertet.

- Eine Transition kann ohne zwischengeschaltete Schritte direkt auf eine andere Transition folgen. Ist eine Transitionsbedingung erfüllt, wird die Transition gesperrt und das nächste Ablaufkettenelement (Schritt oder Transition) wird aktiviert. Bei Laufzeit wird zwischen zwei aufeinanderfolgenden Transitionen ein "Dummyschritt" eingefügt (entsprechend der IEC-Norm). Dieser Dummyschritt führt keine logische Funktion aus, benötigt aber für Aktivierung und Deaktivierung wie jeder normale Schritt auch zwei E/A-Zyklen.
- Die logischen Aktionselemente wurden durch einen Transitionsmerker (TRANS) erweitert, der die Ausführung aller Befehle innerhalb eines Schrittes beendet, alle Bewegungen anhält, den Schritt deaktiviert, und eine sofortige Transition zu einer Marke veranlaßt, die den gleichen Bezeichner wie der Transitionsmerker besitzt.
- Mit Makroschrittkästchen kann eine komplette Ablaufkette in einen Ablaufketten-Makroschritt eingebunden werden. Bei Aktivierung des Makroschritt wird mit der Bearbeitung der eingebundenen Ablaufkette am Startschritt begonnen. Der Makroschritt ist abgeschlossen, wenn die eingebundene Ablaufkette den Endeschritt bearbeitet.
- Aktionen können mit Makroschritten verbunden werden. Sie werden ausgeführt, solange der Makroschritt aktiv ist.

Erstellen von Ablaufplänen

Eine Ablaufkette erstellen

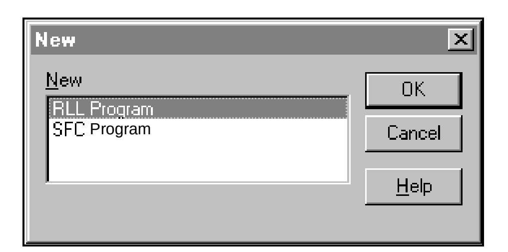
Nachdem Sie den **Programmeditor** gestartet haben, können Sie eine neue Ablaufkette erstellen oder eine bestehende bearbeiten.

Eine neue Ablaufkette erstellen:

1. Klicken Sie auf *Neue Datei* in der Symbolleiste *Programmeditor*.

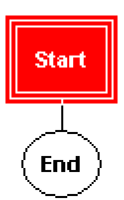
   

   Hierauf erscheint das Menü "Neues Programm".

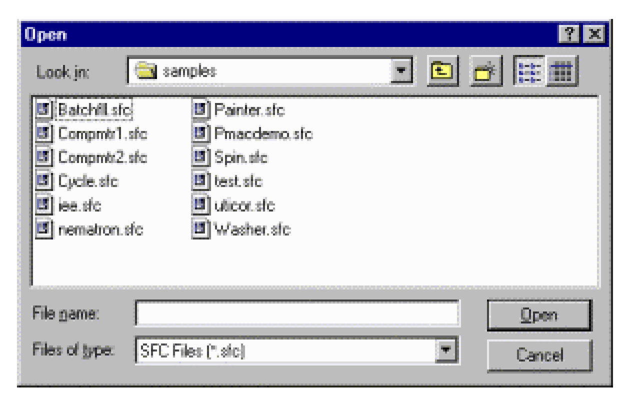
2. Wählen Sie *Ablaufkette* und klicken dann auf *OK*.

   Eine neue AS-Datei mit einem Anfangsschritt und einem Endeschritt erscheint.

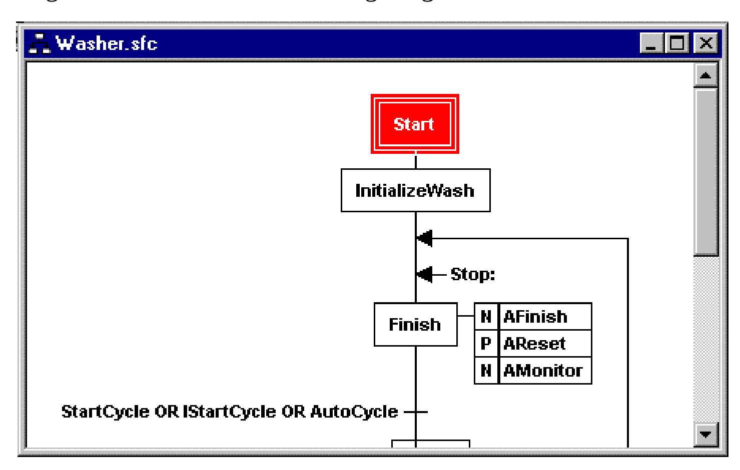


Eine bestehende Ablaufkette bearbeiten:

1. Klicken Sie auf das Werkzeug *Datei öffnen* in der Symbolleiste *Programmeditor*. Hierauf wird die Liste vorhandener Programme angezeigt.



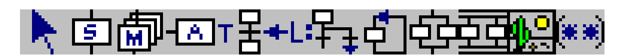
2. Klicken Sie auf das Programm, um es zu öffnen. Es kann sein, daß Sie zunächst den Ablaufkettentyp im Feld **Dateien vom Typ** einstellen müssen. Die ausgewählte Ablaufkette wird angezeigt.



3. Beginnen Sie mit der Bearbeitung der Programmelemente.

Verwendung von AS-Symbolleiste und Menüleiste

Die AS-Symbolleiste enthält alle Werkzeuge, die Sie zum Erstellen einer Ablaufkette benötigen.



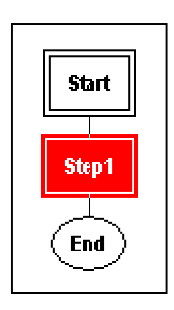
Symbol	Option in Symbolleiste	Funktion
	—	Auswahl von Programmelementen.
	Einfügen/Schritt	Fügt dem Programm einen Schritt hinzu.
	Einfügen/Makroschritt	Fügt dem Programm einen Makroschritt hinzu.
	Einfügen/Aktion	Fügt dem Programm eine Aktion hinzu.
	Einfügen/Transition	Fügt dem Programm eine Transition hinzu.
	Einfügen/Marke	Fügt dem Programm eine Marke hinzu.
	Einfügen/Sprung	Fügt dem Programm einen Sprung hinzu.
	Einfügen/Schleife	Fügt dem Programm eine Schleife hinzu.
	Einfügen/Auswahlverzweigung	Fügt dem Programm eine Auswahlverzweigung hinzu.
	Einfügen/Simultanverzweigung	Fügt dem Programm eine Simultanverzweigung hinzu.
	Einfügen/ Neues Anwendungssymbol ...	Fügt dem Programm einen vordefinierten Programmschritt aus der Bibliothek hinzu.
	Einfügen/Kommentar	Gestattet es, dem Programm Kommentare hinzuzufügen.

Mit Schritten arbeiten

Ein Schritt stellt einen Zustand dar, in dem das Systemverhalten einem Regelsatz folgt, der von den mit dem Schritt verknüpften Aktionen und Funktionen definiert wird. Nachdem Sie einem Programm einen Schritt hinzugefügt haben, müssen Sie ihn konfigurieren.

Einen Schritt hinzufügen

1. Klicken Sie in der *AS-Symbolleiste* auf das Symbol *Schrittwerkzeug*. Der Cursor wird zum Schrittcursor.
2. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie den neuen Schritt einsetzen wollen, und klicken Sie. Der neue Schritt erscheint im Programm an der angegebenen Stelle.



3. Konfigurieren Sie den Schritt.

Einen Schritt konfigurieren

Zum Konfigurieren eines Schritts müssen Sie:

- die Programmiersprache angeben, die Sie für den Schritt verwenden wollen;
- die Daten auswählen, die Sie in dem Schritt anzeigen wollen (z.B. Schrittname, Schrittbeschreibung, Prozeßbefehle, oder Symbol).
- Geben Sie den Programmcode für den Schritt ein oder verknüpfen Sie eine Datei, die den Programmcode enthält, mit dem Schritt.

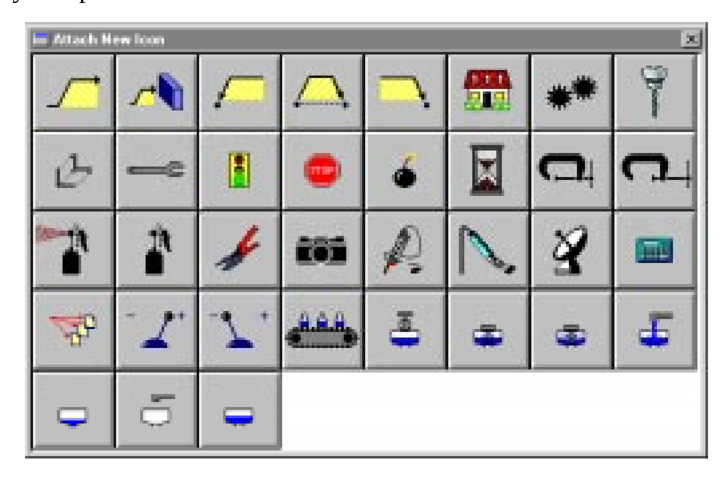
1. Klicken Sie zuerst auf das *Auswahlwerkzeug* und doppelklicken Sie anschließend auf den Schritt. Hierauf erscheint die Dialogbox *Schritt editieren*.
2. Geben Sie die für Konfiguration und Programmierung des Schritts erforderlichen Daten ein. Verwenden Sie dabei die Informationen aus der

nachstehenden Tabelle. Klicken Sie dann auf **OK**, um Ihre Änderungen zu speichern und schließen Sie die Dialogbox.

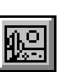
Um ...	**... müssen Sie ...**
die RS-274D Programmiersprache auszuwählen,	auf RS-274D klicken.
strukturierten Text als Programmiersprache auszuwählen,	auf strukturierten Text klicken.
dem Schritt Programmcode hinzuzufügen,	1. auf Prozeßbefehle klicken, 2. den Programmcode eingeben oder eine Datei verknüpfen.
den Schritt mit einer Datei zu verknüpfen, die den Programmcode enthält,	auf "Dateiverknüpfung" klicken.
die verknüpfte Programmcode-Datei für den Schritt zu öffnen und in einem Editor anzuzeigen,	auf "verknüpfte Datei bearbeiten" klicken.
den Symbolmanager aufzurufen,	auf "Symbolmanager-Anzeige" klicken.
den Schrittnamen im Schritt anzuzeigen,	1. auf "Schrittnamen" klicken, 2. einen Schrittnamen in der Editierbox eingeben.
den Code (strukturierten Text oder Bewegungssteuerung) im Schritt anzuzeigen,	1. auf "Bewegungs-/Prozeßbefehle" klicken, 2. den Code eingeben oder eine Datei verknüpfen. Der gesamte Programmiercode für den Schritt erscheint im Schritt. Bei einer langen Reihe von Befehlen kann dies die Größe des Schrittes beträchtlich erweitern.
die Schrittbezeichnung im Schritt anzuzeigen,	1. auf "Schrittbeschreibung" klicken, 2. auf "Beschreibung bearbeiten" klicken, 3. eine Schrittbeschreibung in der Box "Beschreibung bearbeiten" eingeben.
die Breite (in Pixel) für die Schrittbeschreibung einzustellen,	im Feld "Breite" die Anzahl Pixel für die Schrittbeschreibung eingeben..
ein Symbol im Schritt anzuzeigen,	auf Symbol klicken.
Anwendungssymbolpalette aufzurufen,	auf Symbol klicken. . . .
ein Symbol aus dem Schritt zu löschen, wenn eines zugewiesen wurde,	auf "Entfernen" klicken.

Einen Schritt als Symbol anzeigen

1. Bearbeiten Sie einen Schritt.
2. Klicken Sie in der Schrittbearbeitungs-Dialogbox auf Symbol. Hierauf erscheint die Symbolpalette:



3. Klicken Sie in der Symbolpalette auf das Symbol und geben Sie dann einen Titel ein, wenn das System Sie dazu auffordert.
4. Klicken Sie auf OK, um zur Schrittbearbeitungs-Dialogbox zurückzukehren. Nach der Bearbeitung des Schritts erscheinen Symbol und Titel im Schrittkästchen.

Einen Anwendungssymbolschritt hinzufügen 

Ein Anwendungssymbolschritt ist ein Schritt, der eine für eine bestimmte Funktion vordefinierte Code-Mustervorlage enthält, sowie ein für die Funktion geeignetes Symbol. Die PC Control Software stellt eine Bibliothek mit mehreren Schritten zur Verfügung. Sie können aber auch Ihre eigenen Schritte erzeugen und in einer Ablaufkette einsetzen.

Einen Anwendungssymbolschritt hinzufügen

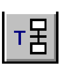
1. Klicken Sie auf das *Werkzeugsymbol* in der AS-Symbolleiste, um die Anwendungssymbole aufzurufen.
2. Legen Sie das Anwendungssymbol in der Ablaufkette ab.
3. Nachdem Sie einen Anwendungssymbolschritt in einer Ablaufkette abgelegt haben, können Sie ihn wie einen normalen Schritt bearbeiten.

Da es sich bei dem Code um eine Mustervorlage handelt, müssen Sie ihn so verändern, daß er Ihrer Anwendung entspricht. Sie können die Schrittoptionen auch andersweilig verändern.

Nachdem Sie einen Anwendungssymbolschritt angepaßt haben, können Sie in in die Bibliothek eingliedern. Bei Bedarf können Sie ihn dann wieder aus der Bibliothek herauskopieren.

Mit Transitionen arbeiten

Eine Transition ist ein graphisches Element in AS-Programmiersprache, das eine einzelne Boolesche Bedingung darstellt, die erfüllt werden muß, ehe die Programmbearbeitung fortgesetzt werden kann.

Eine Kontaktplantransition hinzufügen 

1. Klicken Sie auf *Bearbeiten* in der Menüleiste. Steht neben *Boolesche Transition* ein Häkchen, klicken Sie darauf, um es zu löschen.
2. Klicken Sie auf das *Transitionswerkzeug* in der AS-Symbolleiste. Der Cursor wird zum Transitionscursor.
3. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie die neue Transition einsetzen wollen, und klicken Sie. Die neue Transition erscheint im Programm an der angegebenen Stelle.

Eine Kontaktplantransition bearbeiten

1. Klicken Sie zuerst auf das *Auswahlwerkzeug* und doppelklicken Sie anschließend auf die Transition. Hierauf erscheint die Dialogbox *Kontaktplantransitionslogik auswählen*.
2. Geben Sie einen Namen für die Transition ein und klicken dann auf *OK*.
   - Eine Kontaktplantransition kann den gleichen Namen wie eine Aktion haben. Die Kontaktplanlogik für Aktionen und für Transitionen hat unterschiedliche Bereiche. Die Verwendung gleicher Namen für Aktion und Transition bedeutet daher nicht, daß bei beiden die gleiche Kontaktplanlogik ausgeführt wird.
   - Ein Fenster erscheint, das einen Kontaktplan-Strompfad mit einer Spule enthält, die den gleichen Namen wie die Transition besitzt.
3. Fügen Sie mit dem Kontaktplaneditor die Kontaktplanelemente in den Strompfad ein. Verwenden Sie die gleichen Regeln für Kontakte, Spulen, Sprünge usw.
4. Speichern Sie Ihre Arbeit, indem Sie auf das Kästchen *Steuerungsmenü* klicken. Klicken Sie dann auf *Schließen*.

Eine Boolesche Transition hinzufügen

1. Klicken Sie auf *Bearbeiten* in der Menüleiste und klicken Sie auf *Boolesche Transition*, um das Häkchen anzuzeigen.

2. Klicken Sie auf das *Transitionswerkzeug* in der AS-Symbolleiste. Der Cursor wird zum Transitionscursor.

3. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie die neue Transition einsetzen wollen, und klicken Sie. Die neue Transition erscheint im Programm an der angegebenen Stelle.

Eine Boolesche Transition bearbeiten

1. Zum Bearbeiten der Transition klicken Sie zunächst auf das *Auswahlwerkzeug* . Doppelklicken Sie dann auf die Transition. Hierauf erscheint die Dialogbox *Transitionslogik editieren*.
2. Geben Sie den Booleschen Code für die Transition ein. Sie können den Code direkt eintippen oder auf die Operations-Schaltflächen klicken und aus den vordefinierten Symbolen auswählen.
3. Klicken Sie dann auf *OK*, um Ihre Änderungen zu speichern und schließen Sie die Dialogbox.
4. Sie können den Symbolmanager aufrufen, um lokale Variable zu konfigurieren und eine Liste aller konfigurierten Variablen anzuzeigen, die Sie in dem Booleschen Ausdruck verwenden können.

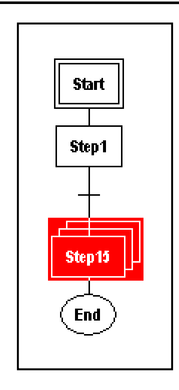
Mit Makroschritten arbeiten

Mit einem Makroschritt können Sie eine andere Ablaufkette aus der aktuell bearbeiteten Ablaufkette aufrufen.

Einen Makroschritt hinzufügen 

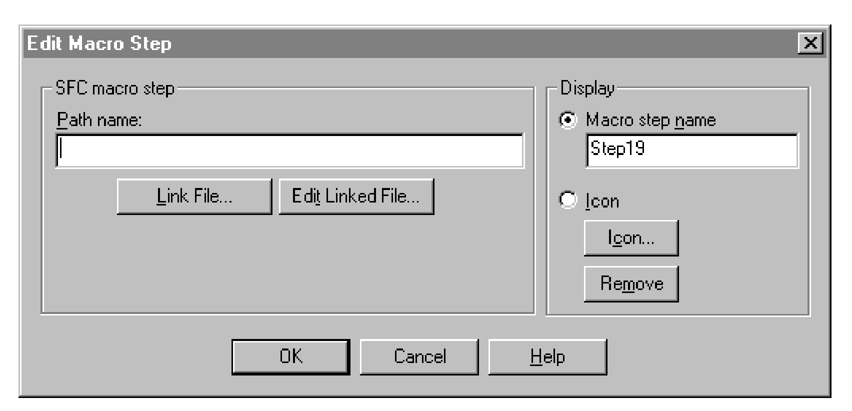
1. Klicken Sie auf das *Makroschritt-Werkzeug* in der AS-Menüleiste. Der Cursor wird zum *Makroschrittwerkzeug-Cursor*.

2. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie den neuen Makroschritt einsetzen wollen, und klicken Sie. Der neue Makroschritt erscheint im Programm an der angegebenen Stelle.



Einen Makroschritt konfigurieren

1. Klicken Sie zuerst auf das *Auswahlwerkzeug* und doppelklicken Sie anschließend auf den Makroschritt. Hierauf erscheint die Dialogbox **Makroschritt editieren**.



2. Geben Sie die entsprechenden Informationen zur Konfiguration des Makroschrittes ein.

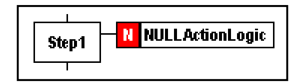
3. Klicken Sie dann auf *OK*, um Ihre Änderungen zu speichern und schließen Sie die Dialogbox.

Mit Aktionen arbeiten

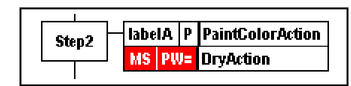
Eine Aktion besteht aus einem oder mehreren Abschnitten Kontaktplancode, die mit einem Schritt oder Makroschritt verknüpft sind. Das System führt eine Aktion aus, wenn der zugehörige Schritt aktiv wird.

Eine Aktion hinzufügen

1. Klicken Sie auf das *Aktionswerkzeug* in der AS-Menüleiste. Der Cursor wird zum Aktionswerkzeug-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Programm (entweder an den Anfang eines Schritts oder an den Anfang eines Makroschritts), an der Sie die neue Aktion einfügen wollen, und klicken Sie auf diese Stelle. Die neue Aktion erscheint im Programm an der angegebenen Stelle.

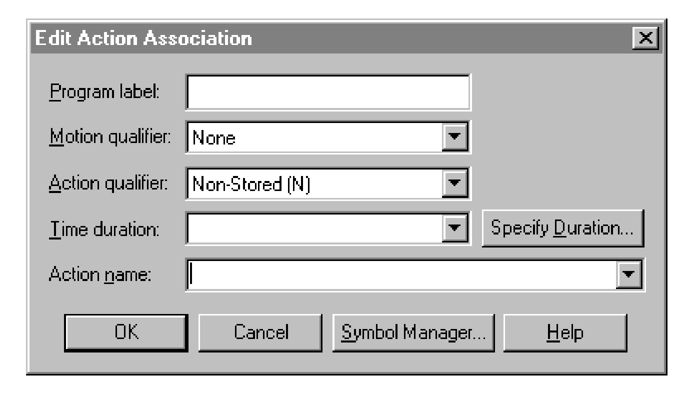


3. Sie können einem Schritt oder Makroschritt mehrere Aktionen hinzufügen. Setzen Sie dazu den Cursor oben auf eine bestehende Aktion.



Eine Aktion konfigurieren

Verwenden Sie zur Konfiguration einer Aktion die Dialogbox "Aktionsverknüpfung bearbeiten".



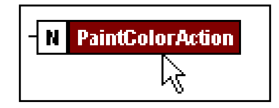
Feld/Schaltfläche	Beschreibung
Programmmarke	(Option) Die Aktion wird erst ausgeführt, nachdem die Marke im Schrittcode aufgetreten ist. Aktion und Programmmarke müssen in der selben Ablaufkette stehen. Zwischen Programmen sind keine Querverweise erlaubt. Die Marken in einer Makroschritt-Ablaufkette können von der Mutter-Ablaufkette aus nicht angesprochen werden. Ebenso können die Marken in der Mutter-Ablaufkette nicht von der Makro-Ablaufkette angesprochen werden. Enthält der Schritt, mit dem die Aktion verknüpft ist, keine Marke, dann wird der Parameter "Programmmarke" ignoriert.
Aktions-Bestimmungszeichen	Gibt ein Aktions-Bestimmungszeichen an.
Zeitdauer	(Nur Aktions-Bestimmungszeichen) gibt die Zeitdauer für begrenzte und Verzögerungs-Bestimmungszeichen an. Verwendet das Aktions-Bestimmungszeichen die Zeitdauer nicht, werden alle für diesen Parameter angegebenen Werte ignoriert.
Zeitdauer angeben	Ruft die Dialogbox "Zeitdauer definieren" auf.
Aktionsbezeichnung	Gibt den Namen der Aktion an.

Eine Aktion bearbeiten

1. Klicken Sie zuerst auf das *Auswahlwerkzeug* und doppelklicken Sie anschließend auf die Aktion. Hierauf erscheint die Dialogbox **Aktionsverknüpfung editieren**. Geben Sie die Konfigurationsdaten der Aktion ein.
2. Klicken Sie auf *OK*, um Ihre Änderungen zu speichern. Das System schließt die Dialogbox und zeigt einen leeren Kontaktplan-Strompfad an, der bereit zur Bearbeitung ist.
3. Geben Sie den Kontaktplancode ein.

Kontaktplan einer Aktion bearbeiten

4. Klicken Sie auf das *Auswahlwerkzeug* .
5. Doppelklicken Sie auf die rechte Hälfte der Aktion (siehe Abbildung).



Das System zeigt den Kontaktplancode für die Aktion an. Geben Sie den Kontaktplancode ein.

Einen Konfigurationsparameter einer Aktion bearbeiten

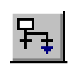
1. Klicken Sie auf das *Auswahlwerkzeug* .
2. Doppelklicken Sie auf die linke Hälfte der Aktion (siehe Abbildung). Das System zeigt die Dialogbox **Aktionsverknüpfung bearbeiten** für die Aktion an.



3. Geben Sie die Informationen zur Konfiguration der Aktion ein.
4. Klicken Sie auf *OK*, um Ihre Änderungen zu speichern.

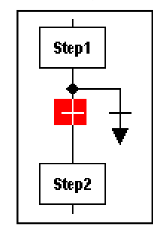
Ablaufketten-Steuerungselemente hinzufügen

Sie können den Programmablauf mit Sprüngen, Schleifen und Verzweigungen steuern.

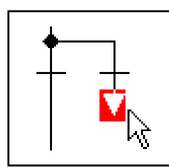
Einen Sprung hinzufügen 

Eine Kombination "Sprung zur Marke" ermöglicht die Fortführung der AS-Bearbeitung an einer beliebigen durch eine Marke gekennzeichneten Stelle.

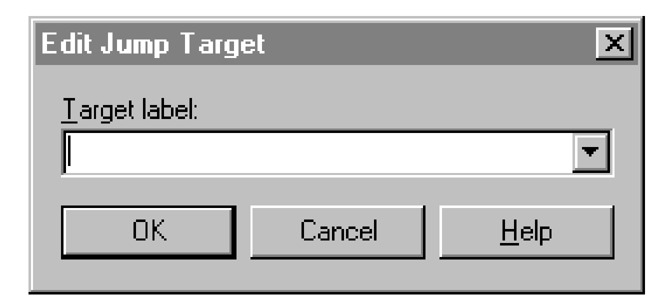
1. Klicken Sie in der Menüleiste auf *Bearbeiten* und wählen Sie *Boolesche Transition*.
2. Klicken Sie auf das *Sprungwerkzeug* in der AS-Menüleiste. Der Cursor wird zum Sprungwerkzeug-Cursor.
3. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie den neuen Sprung einsetzen wollen, und klicken Sie. An der angegebenen Stelle im Programm erscheinen der neue Sprung und zwei Transitionen.



4. Zur Bearbeitung des Sprungs klicken Sie zunächst auf das *Auswahlwerkzeug* . Doppelklicken Sie anschließend auf den Pfeil des Sprungs.



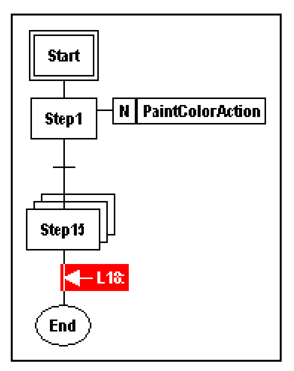
Hierauf erscheint die Dialogbox **Sprungziel editieren**.



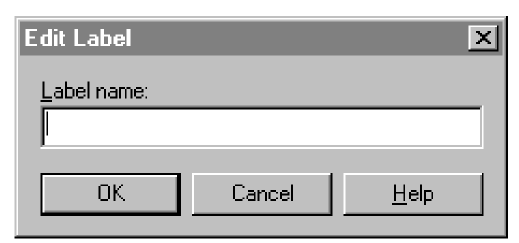
5. Geben Sie die Marke ein, zu der der Sprung den Programmfluß weiterleiten soll. Klicken Sie dann auf *OK*, um Ihre Änderungen zu speichern und schließen Sie die Dialogbox.
6. Bearbeiten Sie die beiden Transitionen.

Eine Marke hinzufügen

1. Klicken Sie auf das *Markenwerkzeug* in der AS-Menüleiste. Der Cursor wird zum Markenwerkzeug-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie die neue Marke einsetzen wollen, und klicken Sie. Die neue Marke erscheint im Programm an der angegebenen Stelle.



3. Zum Bearbeiten klicken Sie zunächst auf das *Auswahlwerkzeug* . Doppelklicken Sie dann auf die Marke. Hierauf erscheint die Dialogbox **Marke editieren**.



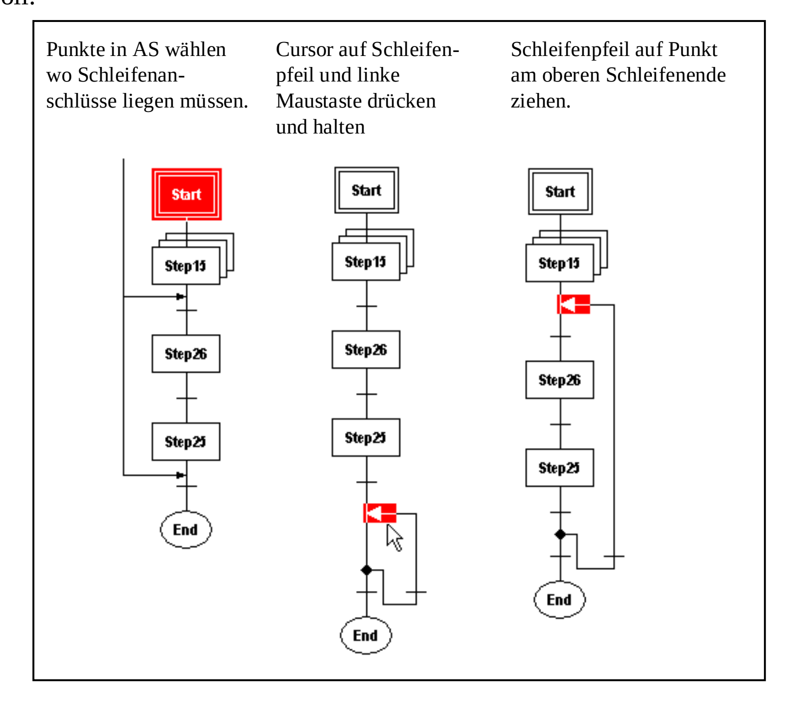
4. Geben Sie eine sinnvolle Marke ein und klicken dann auf *OK*, um Ihre Änderungen zu speichern. Schließen Sie die Dialogbox.

Schleife hinzufügen

Mit einer Schleife können Sie die AS-Programmausführung wieder zurück einer vorangehenden Stelle im Programm führen und so eine Reihe von Schritten wiederholen.

1. Klicken Sie auf das *Schleifenwerkzeug* in der AS-Menüleiste. Der Cursor wird zum Schleifenwerkzeug-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie das untere Ende der Schleife einsetzen wollen, und klicken Sie. Die Schleife erscheint im Programm an der angegebenen Stelle.
3. Klicken Sie auf das *Auswahlwerkzeug* und setzen Sie dann den Cursor über den Schleifenpfeil.

4. Drücken Sie die linke Maustaste und halten Sie sie gedrückt, während Sie den Schleifenpfeil auf den Punkt bewegen, an dem das obere Schleifenende liegen soll.



5. Bearbeiten Sie die Schleifentransitionen.

Eine Schleife verschieben

Oberen Teil der Schleife ziehen:

1. Wählen Sie das Auswahlwerkzeug aus der *Kontaktplan-Symbolleiste* oder der *AS-Symbolleiste*.
2. Setzen Sie den Cursor auf den oberen Pfeil der Schleife. Drücken Sie die linke Maustaste und halten Sie sie gedrückt. Ziehen Sie den Schleifenpfeil auf den gewünschten Punkt (der Cursor verändert seine Gestalt in einen Schleifenpfeil-Cursor).
3. Lassen Sie die linke Maustaste los, um den Schleifenpfeil am gewünschten Punkt abzulegen. Drücken Sie die Taste <ESCAPE>, um den Ziehvorgang rückgängig zu machen.

Der Zielpunkt des Schleifenpfeils liegt ...	**...dann...**
auf der gleichen Verzweigung und nicht unterhalb der Schleifentransition, mit der dieser Schleifenpfeil verbunden ist, nicht in einer eingebetteten Schleife, nicht außerhalb von verschachtelten Schleifen,	liegt der gezogene Schleifenpfeil an der Ablegeposition und die Schleifentransition bleibt an ihrer aktuellen Position.
auf einer anderen Verzweigung, unterhalb der Schleifentransition, mit der dieser Schleifenpfeil verbunden ist, oder außerhalb von verschachtelten Schleifen,	werden die ganze Schleife mit dem gesamten Inhalt zur Zielposition verschoben.
in einer eingebetteten Schleife,	ist ein Ziehen nicht möglich.

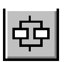
Unteren Teil der Schleife ziehen:

1. Setzen Sie den Cursor über die Schleifentransition. Drücken Sie die linke Maustaste und halten Sie sie gedrückt. Ziehen Sie die Schleifentransition auf den gewünschten Punkt (der Cursor verändert seine Gestalt in einen Schleifentransitions-Cursor).
2. Lassen Sie die linke Maustaste los, um die Schleifentransition am gewünschten Punkt abzulegen.

Drücken Sie die Taste <ESCAPE>, um den Ziehvorgang rückgängig zu machen.

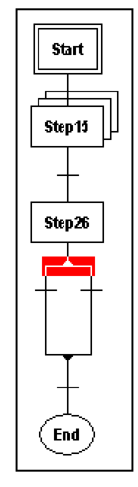
Der Zielpunkt der Schleifentransition liegt ...	**... dann ...**
auf der gleichen Verzweigung und nicht oberhalb von einem der mit der Schleifentransition verbundenen Schleifenpfeile, nicht in einer eingebetteten Schleife,	liegt die gezogene Schleifentransition an der Ablegeposition und die Schleifentransition bleibt an ihrer aktuellen Position.

Der Zielpunkt der Schleifentransition liegt ...	... dann ...
nicht außerhalb von verschachtelten Schleifen,	
auf einer anderen Verzweigung, oberhalb eines der mit der Schleifentransition verbundenen Schleifenpfeils, oder außerhalb eines verschachtelten Zweiges,	werden die ganze Schleife mit dem gesamten Inhalt zur Zielposition verschoben.
in einer eingebetteten Schleife,	ist Ziehen nicht möglich und die Schleife kehrt zu ihrer ursprünglichen Position zurück.

Eine Auswahlverzweigung hinzufügen 

Mit einer Auswahlverzweigung kann die AS-Programmausführung über einen von mehreren Pfade gelenkt werden.

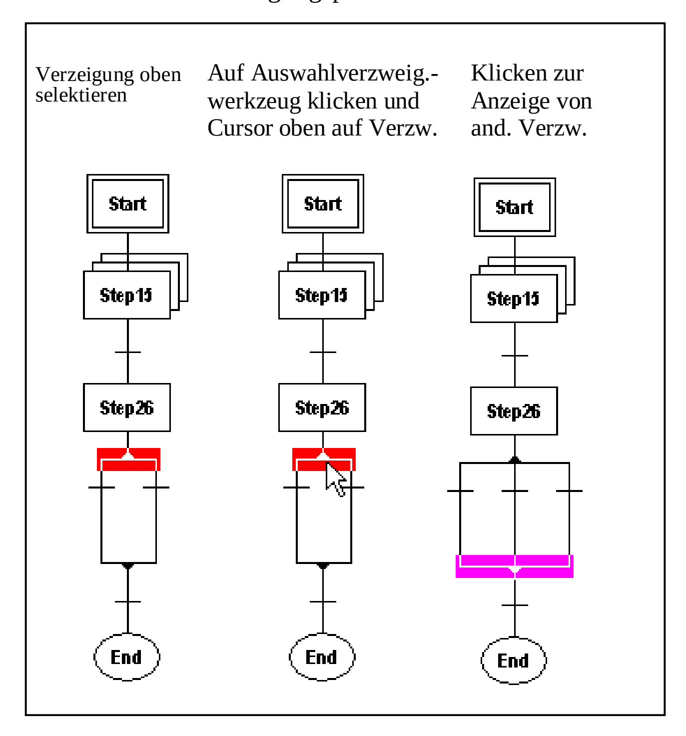
1. Klicken Sie auf *Bearbeiten* in der Menüleiste und wählen dann als den mit der Verzweigung zu benutzenden Transitionstyp *Boolesche Transition* aus (Kontaktplan oder Boolesche).
2. Klicken Sie auf das *Auswahlverzweigungs-Werkzeug* in der AS-Menüleiste. Der Cursor wird zum Auswahlverzweigungswerkzeug-Cursor.
3. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie die Auswahlverzweigung einsetzen wollen, und klicken Sie. Die neue Auswahlverzweigung erscheint im Programm an der angegebenen Stelle.



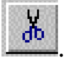
4. Bearbeiten Sie die beiden Transitionen.

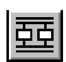
Einen weiteren Pfad hinzufügen:

1. Selektieren Sie den oberen Teil der Auswahlverzweigung, ehe Sie eine weitere Auswahlverzweigung hinzufügen.
2. Setzen Sie den Cursor oben auf die Verzweigung und klicken Sie. Hierauf erscheint ein weiterer Verzweigungspfad.



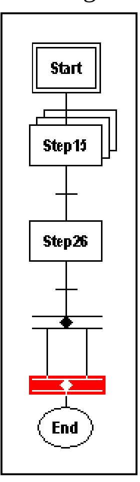
Einen Pfad löschen:

1. Selektieren Sie den Pfad, den Sie löschen wollen.
2. Klicken Sie auf das Ausschneidewerkzeug .

Eine Simultanverzweigung hinzufügen 

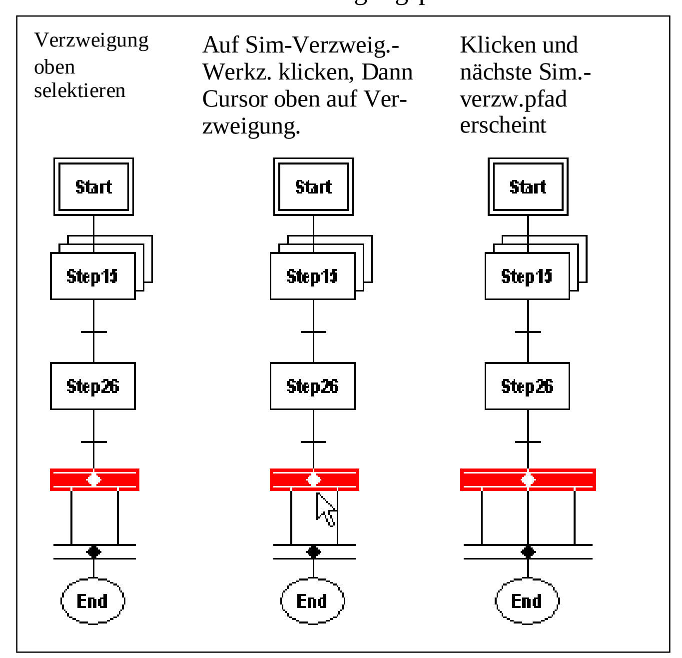
Mit einer Simultanverzweigung kann die AS-Programmausführung über zwei oder mehrere Pfade gelenkt werden. Damit die Programmausführung über die Simultanverzweigung hinausgeht muß die Bearbeitung auf allen Einzelpfaden abgeschlossen sein.

1. Klicken Sie auf das *Simultanverzweigungs-Werkzeug* in der AS-Menüleiste. Der Cursor wird zum Simultanverzweigungswerkzeug-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie die Simultanverzweigung einsetzen wollen, und klicken Sie. Die Simultanverzweigung erscheint im Programm an der angegebenen Stelle.



Einen weiteren Pfad hinzufügen:

1. Selektieren Sie den oberen Teil der Simultanverzweigung, ehe Sie eine weitere hinzufügen.
2. Setzen Sie den Cursor oben auf die Simultanverzweigung und klicken Sie. Hierauf erscheint ein weiterer Verzweigungspfad.



Einen Pfad löschen:

1. Selektieren Sie den Pfad, den Sie löschen wollen.
2. Klicken Sie auf das *Ausschneidewerkzeug* .

Eine vollständige Simultanverzweigung löschen:

1. Selektieren Sie entweder den oberen oder den unteren Teil der Simultanverzweigung.
2. Klicken Sie auf das *Ausschneidewerkzeug*.

Richtlinien für den Einsatz von Simultanverzweigungen

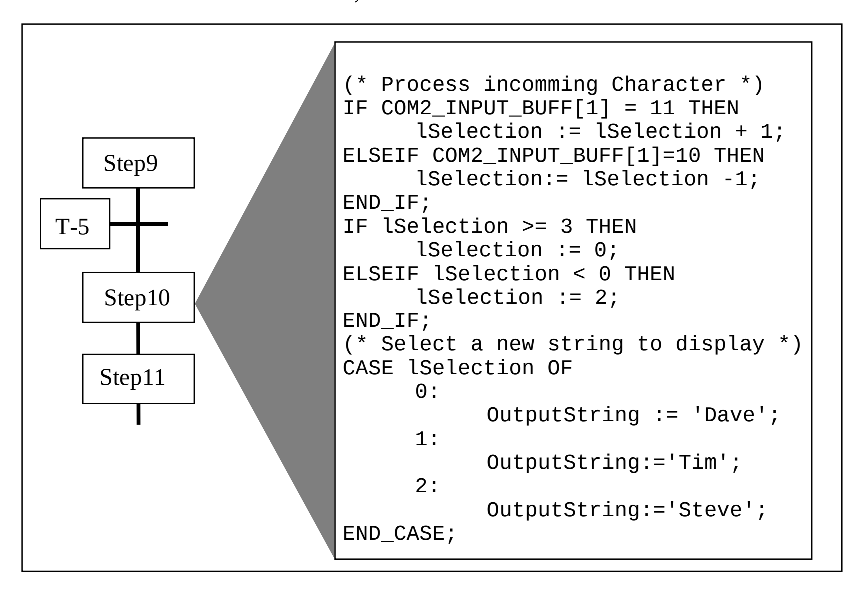
Beachten Sie die folgenden Richtlinien, wenn Sie eine Simultanverzweigung erstellen.

Um eine saubere Verzweigung sicherzustellen, dürfen Sie Marken nicht in folgender Weise verwenden:

- um auf einen Punkt außerhalb einer Simultanverzweigung zu springen.
- um in eine Simultanverzweigung hinein zu springen.
- um zu einem anderen Pfad innerhalb einer Simultanverzweigung zu springen.

Strukturierten Text in eine Ablaufkette integrieren

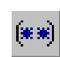
Die Programmiersprache "strukturierter Text" ist ein der IEC-1131-3 entsprechender Satz Anweisungen auf Textbasis, der zur einfachen Erstellung mathematischer und logischer Operationen geschaffen wurde. Bei der Erstellung des Anwendercodes zu einem Ablaufketten-Schritt können Sie wahlweise strukturierten Textcode verwenden (siehe nachstehende Abbildung). Wird die Ablaufkette bearbeitet, dann wird der in den einzelnen Schritten eingegliederte strukturierte Textcode verarbeitet, sobald der Schritt aktiv wird.



Bei der Programmentwicklung müssen Sie diese Punkte beachten.

- Der Rumpf des strukturierten Textprogrammcodes selbst besteht aus Ausdrücken, Anweisungen und Funktionen. Im strukturierten Text zulässige Operatoren und Datentypen werden im Abschnitt "Ausdrücke in strukturiertem Text" beschrieben.
- Führen Sie alle im Programm benötigten Symbolvereinbarungen vom Symbolmanager aus durch.
- Sie können Symbole über eine oder mehrere Aktionen initialisieren, die Sie mit dem Schritt verknüpfen können.
- Stelen Sie sicher, daß Sie keine der PC Control Schlüsselwörter verwenden.

Dokumentieren einer Ablaufkette

Programmkommentare hinzufügen 

Ein Programmkommentar kann aus jeder sinnvollen Beschreibung bestehen, die Sie zu neben einem Programmelement anzeigen wollen. Sie können einstellen, ob das System die Kommentare anzeigt oder verbirgt.

1. Klicken Sie in der AS-Menüleiste auf das *Kommentarwerkzeug*. Der Cursor wird zum Programmkommentarwerkzeug-Cursor.
2. Setzen Sie den Cursor auf die Stelle im Programm, an der Sie den Kommentar einsetzen wollen, und klicken Sie. Hierauf erscheint die Dialogbox *Programmkommentare*.
3. Geben Sie den Kommentar ein und klicken dann auf *OK*. Der Kommentar erscheint im Programm an der angegebenen Stelle.

Programmkommentare bearbeiten

1. Doppelklicken Sie auf den Kommentar, den Sie bearbeiten wollen. Hierauf erscheint die Dialogbox **Programmkommentare**.
2. Bearbeiten Sie den Kommentar entsprechend.

Einen Kommentar anzeigen

Klicken Sie auf *Anzeigen* und wählen Sie *Programmkommentare* oder klicken Sie auf die Schaltfläche "Kommentare anzeigen" in der Editor-Symbolleiste.

Betriebsart "Kommentare anzeigen" deaktivieren:

Klicken Sie auf *Anzeigen* und wählen Sie *Programmkommentare* oder klicken Sie erneut auf die Schaltfläche "Kommentare anzeigen".

Hinweis

> Die Betriebsart "Kommentare anzeigen" wird für alle Dateien aktiviert oder deaktiviert.

Bei aktivierter Betriebsart "Kommentare anzeigen" steht im Anzeigemenü ein Häkchen neben dem Befehl "Programmkommentar" und in der Bearbeitungs-Symbolleiste wird die Schaltfläche "Kommentare anzeigen" als gedrückt angezeigt.

4

Abschnitt 4: Programmierung in strukturiertem Text

Übersicht

Strukturierter Text ist eine textstrukturierte Programmiersprache entsprechend IEC 1131-3. Diese Sprache ist bequem für Personen mit Erfahrung in strukturiertem BASIC, Pascal, C oder anderen höheren Programmiersprachen.

Zur Erstellung von eigenständigen Programmen in strukturiertem Text verwenden Sie den Editor für strukturierten Text. Dieser Editor enthält typische Textbearbeitungsfunktionen, wie zum Beispiel Ausschneiden, Kopieren, Einfügen, Suchen und Ersetzen. Er besitzt auch Werkzeuge und Befehle zum automatischen Einfügen von Anweisungskonstrukten, wie zum Beispiel IF und CASE Auswahlbefehle und FOR, REPEAT und WHILE Schleifen.

Sie können auch Befehle in strukturiertem Text in einen Ablaufketten-Schritt eingliedern. Eine patentierte Erweiterung zur Ablaufsprache gestattet die Integration von strukturiertem Text in einen Ablaufketten-Schritt. Bei der Erstellung des Anwendercodes zu einem Ablaufketten-Schritt können Sie wahlweise strukturierten Textcode verwenden. Wird die Ablaufkette bearbeitet, dann wird der in den einzelnen Schritten eingegliederte strukturierte Textcode verarbeitet, sobald der Schritt aktiv wird. Mit Ausnahme von bestimmten Funktionen arbeiten der eigenständige Editor und der Editor für strukturierten Text in Ablaufketten-Schritten gleich.

Dieser Abschnitt informiert über die Verwendung des Editors für strukturierten Text. Es wird angenommen, daß Sie die allgemeine Arbeitsweise eines Programmeditors kennen und mit der strukturierten Textsprache etwas vertraut sind.

Hinweis

Eigenständige Programme in strukturiertem Text laufen einmal ab und werden dann verlassen.

Öffnen eines Dokuments in strukturiertem Text

Zum Öffnen eines bestehendes Dokuments wählen Sie *Editor öffnen* im Programmeditormenü *Datei* und selektieren das Dokument mit der daraufhin erscheinenden Dialogbox *Öffnen*.

Zum Öffnen eines neuen Dokuments wählen Sie *Editor Neu* im Programmeditormenü *Datei* und wählen *strukturiertes Textdokument* aus der daraufhin erscheinenden Dialogbox *Neu*.

Strukturierten Text in einem Ablaufketten-Schritt bearbeiten

Um strukturierten Text in einem bestehenden Ablaufketten-Schritt zu bearbeiten, doppelklicken Sie auf den Schritt oder selektieren Sie den Schritt und wählen dann *Element bearbeiten* aus dem Menü *Bearbeiten* oder *ST oder Makroschritt bearbeiten* aus dem Kontextmenü. Stellen Sie sicher, daß *Schritteigenschaften* auf *strukturierten Text* eingestellt ist.

Anweisungen eingeben

Eingabe von Hand

Anweisungen können eingegeben werden, indem Anweisung oder Funktionsaufruf und die zugehörigen Parameter über die Tastatur eingegeben werden. Beachten Sie dabei die in "Sprachübersicht" aufgeführte Anweisungssyntax sowie die in "Sprachreferenz" angegebenen Syntax der Funktionsaufrufe.

Zubehörleiste

Die Zubehörleiste zeigt Befehle in graphischer Form an. Sie bietet eine alternative Methode der Befehlseingabe. Die Zubehörleiste erscheint nur, wenn Sie über *Zubehörleiste* im Menü *Anzeige* freigegeben wurde. Die nachstehende Abbidlung zeigt die Funktionen der Zubehörleiste. Die frei bewegliche Zubehörleiste kann losgelöst und nach Bedarf plaziert werden.

Comment
Assignment
Label
Break
Exit
Include
Scan
If
Case
For Loop
Repeat Loop
While Loop

Menü "ST-Anweisungen einfügen"

Bei aktivem Editor für strukturierten Text enthält das Programmeditormenü *Bearbeiten* ein Element *ST-Anweisung einfügen*, das strukturierte Textanweisungen auflistet. Wählen Sie die benötigte Anweisung und sie wird automatisch an der Cursorposition in richtiger Syntax eingefügt. Ersetzen Sie die Parameter und Ausdrücke nach Bedarf. Wahlweise benutzbare Teile einer Anweisung sollten entfernt werden, wenn sie nicht verwendet werden.

Menü "ST-Anweisungen einfügen"

Ist der Editor für strukturierten Text aktiv, enthält das Programmeditormenü *Bearbeiten* ein Element *ST-Funktionsaufrufe einfügen*, das die Standardfunktionen aufführt, die ím Zusammenhang mit der strukturierten Textsprache verwendet werden können. Wählen Sie die benötigte Funktion und sie wird automatisch an der Cursorposition in richtiger Syntax eingefügt. Ersetzen Sie alle Parameter mit denen, die Sie im Symbolmanager definiert haben.

Weitere Information zur Verwendung von Funktionen und Funktionsblöcken in der strukturierten Textsprache finden Sie in "Sprachübersicht**”**.



Strukturierten Text bearbeiten

Der Editor für strukturierten Text unterstützt die allgemeinen Editorfunktionen, wie zum Beispiel Ausschneiden, Kopieren, Einfügen, Suchen und Ersetzen. Diese Befehle finden Sie unter *Bearbeiten* und im Kontextmenü.

Sprachübersicht

Ausdrücke

Ein Ausdruck ist definiert als eine Kombination von Operatoren (mathematisch, logisch, relational) und Operanden (Konstanten, Symbole, Literalwerte, sonstige Ausdrücke), die ausgewertet werden können und ein Ergebnis in einem der unterstützten Datentypen (z.B. ganzzahlig, reell, usw.) liefern.

Operatoren

In der nachstehenden Tabelle sind die Operatoren aufgeführt, die Sie in einem Ausdruck verwenden können. Die Priorität legt fest, in welcher Reihenfolge die Operatoren im Ausdruck bearbeitet werden. Der Operator mit der höchsten Priorität wird zuerst angewandt, anschließend der Operator mit der zweithöchsten Priorität. Operatoren mit gleicher Priorität werden von links nach rechts ausgewertet.

Operator	Symbol	Priorität
Strukturindex	.	1 (höchste)
Feldindex	[ ]	1
Pointer Referenzierung	&	2
Pointer Dereferenzierung	*	2
Runde Klammer	( )	3
Funktionsauswertung	Bezeichner (Argumentliste) z.B. LN (A), ABS (X)	3
Potenzierung	**, POW	4
Negierung	-	5
Komplement	NOT	5
Multiplikation	*	6
Division	/	6
Modulo	MOD	6
Addition	+	7
Subtraktion	-	7
Vergleich	<, >, <=, >=	8
Gleichheit	=	9
Ungleichheit	<>	9
Booelesches/bitweises UND	AND	10
Booelesches/bitweises Exklusiv-ODER	XOR	11
Booelesches/bitweises ODER	OR	12 (niedrigste)

Diese Symbole besitzen folgende Funktionen:

- := weist einen Ausdruck einem Symbol zu
- ; wird benötigt, um das Ende einer Anweisung zu bezeichnen
- [ ] wird zur Feldindizierung benutzt, wenn der Feldindex eine ganze Zahl ist. Der nachstehende Ausdruck setzt zum Beispiel das ***i***-te Element eines Feldes auf den Wert j+10: intarray[i] = j + 10;
- :: wird zur Kennzeichnung einer Marke verwendet. Zum Beispiel: spray_on:: Nach einer Marke muß in der gleichen Zeile eine Anweisung stehen.
- (* *) kennzeichnet einen Kommentar. Zum Beispiel: (*dies ist ein Kommentar.*)

Pointer-Operatoren

Vorsicht

Pointer sollten nur von Fachleuten eingesetzt werden. Falsche Verwendung kann zu unvorhersehbarem Betriebsverhalten und großen Problemen beim Austesten führen.

Strukturierter Text besitzt zwei Pointeroperatoren: Den Pointer Referenzierung & Operator und den Pointer Dereferenzierung * Operator. Diese Operatoren werden zur indirekten Adressierung eingesetzt. Erfahrene Anwender finden weitere Informationen hierzu in Anhang E.

Syntax von strukturiertem Text

Wenn Sie zum Einfügen von strukturierten Textanweisungen und Funktionen die integrierten Editorwerkzeuge verwenden (ST Zubehörleiste und Einfügemenüs), dann wird die richtige Syntax automatisch eingetragen. Schlüsselwörter erscheinen ganz in Großbuchstaben, vom Anwender austauschbare Ausdrücke und Parameter werden gemischt in Groß- und Kleinschreibung dargestellt, und Optionen erscheinen in eckigen Klammern []. Weitere Angaben zur Verwendung von strukturiertem Text, Funktionen und Funktionsblöcken finden Sie in der **Sprachreferenz**.

Hinweise:

- Anweise in strukturiertem Text müssen mit einem Semikolon (;) enden.
- Symbole in strukturiertem Text müssen im Symbolmanager vereinbart werden.

Zuweisungsbefehl

Der Zuweisungsbefehl ersetzt den Wert einer Variablen durch das Ergebnis eines Ausdrucks (vom gleichen Datentyp).

Format

```
Variable := Ausdruck;
```

mit:
Variable ist ein Symbol, Feld, Feldelement, usw.
Ausdruck ist ein einzelner Wert, Ausdruck, oder komplexer Ausdruck.

Beispiele

Boolesche Zuweisungsbefehle:

```
VarBool1 := TRUE;

VarBool2 := (val <= 75);
```

Feldelement-Zuweisung:

```
Array_1[13] := (RealA /RealB)* PI;
```

Zeichenfolgen-Zuweisung. Zeichenfolgen-Literale müssen in einzelne Anführungszeichen eingeschlossen werden.

```
String_Val := 'dies ist eine Stringkonstante';
```

Funktionswert-Zuweisung:

```
Ergebnis := SQRT(2);
```

Funktionsblock-Wertzuweisung:

Liegt die Instanz CTU1 des Zählerfunktionsblocks in einem Kontaktplanprogramm, dann erhält die folgende Zuweisung im strukturierten Textprogramm den aktuellen Wert des Zählers.

```
CurrentValue:=CTU1.CV;
```

Pointer

- Ist pVar1 ein Pointersymbol, dann ist pVar1 die Adresse des Datenwerts X zugewiesen.

  ```
  pVar1 := & X;
  ```

- Ist pVar1 ein Pointersymbol, dann ist Y der in Var1 enthaltene Wert zugewiesen, da pVar1 die Speicheradresse von Var1 enthält.

  ```
  Y := * pVar1;
  ```

- Ist pVar1 ein Pointersymbol, dann ist Var1 der in Y enthaltene Wert zugewiesen.

  * pVar1 := Y;

BREAK-Anweisung

Die Anweisung BREAK stoppt die Programmausführung, wenn Austesten freigegeben ist (das Programm wurde mit dem Befehl *Lauf mit Austesten* gestartet).

Format

```
BREAK;
```

Beispiel

Ist Austesten freigegeben, stoppt die Programmausführung an der Anweisung BREAK. Die Anweisungen in StatementList2 sowie die darauffolgenden Programmanweisungen können im Einzelschrittbetrieb ausgeführt werden.

```
StatementList1;
BREAK;
StatementList2;
```

CASE-Anweisung

Mit dem Konstrukt CASE ist eine bedingte Ausführung von Anweisungslisten im Multiple-Choice-Verfahren möglich. Es wird bedingt eine von mehreren Anweisungslisten ausgeführt, in der die Bedingung durch den Wert einer ganzzahligen Variablen bestimmt wird.

Format

```
CASE  IntExpression OF
      Int:                    (*Singular*)
      StatementList;
      Int,Int,Int:            (*aufgezählt*)
      StatementList;
      Int..Int:               (*Bereich*)
      StatementList;
[ELSE                         (*Option*)
      StatementList;]
END_CASE;
```

mit:

IntExpression Eine Variable oder ein Ausdruck vom Datentyp ANY_INT.

Int eine ganze Zahl. Es können keine oder mehr der singulären, aufgezählten, oder Bereichsform verwendet werden.

StatementList Null oder mehr strukturierte Textanweisungen.

Arbeitsweise

Die Werte Int werden mit IntExpression verglichen. Die Anweisungsliste StatementList, die auf den ersten Wert Int folgt, der mit IntExpression übereinstimmt, wird ausgeführt. Stimmt kein Wert Int mit IntExpression überein, wird die auf ELSE folgende StatementList ausgeführt; sonst wird keine StatementList ausgeführt. Der ELSE-Teil des CASE-Konstrukts ist optional.

Beispiel

Der nachstehende Codeteil teilt einer String-Variablen einen Wert zu.

```
CASE ColorSelection OF
      0:
            ColorString := 'Rot';
      1:
            ColorString:='Gelb';
      2,3,4:
            ColorString:='Grün';
      5..9:
            ColorString:='Blau';
ELSE
            ColorString:='Violett';
END_CASE;
```

Kommentare

Über Kommentare können Sie Ihren Programmcode mit hilfreichen Anmerkungen versehen, die die Arbeitsweise des Programms dokumentieren. Der Compiler ignoriert alle Zeichen, die zwischen einem Paar Sternchen (*…*) stehen. Eiin Kommentar kann am Ende einer Codezeile oder auf einer eigenen Zeile stehen.

Format

```
(*frei wählbarer Text*)
```

Beispiel

```
Result := SQRT(x);      (*Verwendet die Wurzelfunktion*)
```

Exit-Anweisung

Mit der Anweisung EXIT können Sie eine Schleife (FOR, WHILE, REPEAT) vorzeitig beenden und verlassen. Die Programmausführung wird mit der auf das Schleifenabschlußelement (END_FOR, END_WHILE, END_REPEAT) folgenden Anweisung fortgesetzt.

Format

```
ConditionForExiting EXIT;
```

mit:

ConditionForExiting Ein Ausdruck der die frühzeitige Beendigung festlegt.

Beispiel

Der folgende Codeteil zeigt die Arbeitsweise der Anweisung EXIT. Wird die Variable number größer als 500, dann wird die FOR-Schleife verlassen und die Programmausführung mit der Anweisung fortgesetzt, die unmittelbar auf END_FOR folgt.

```
number:=1

FOR counter := 1 TO 100 DO
      number := number * counter;
      IF number > 500 THEN EXIT;
END_FOR;
```

IF-Anweisung

Mit dem IF-Konstrukt kann eine Anweisungsliste bedingt ausgeführt werden. Die Bedingung wird durch das Ergebnis eines Booleschen Ausdrucks bestimmt. Ein IF-Konstrukt besitzt zwei optionale Teile: Die eine Option ermöglicht die bedingte Ausführung einer durch einen zweiten Booleschen Ausdruck festgelegten alternativen Anweisungsliste. Eine weitere Option ermöglicht die unbedingte Ausführung einer dritten Anweisungsliste, wenn keine Bedingung erfüllt wird.

Ist keiner der Boolescher Ausdrücke TRUE und haben Sie eine ELSE-Anweisung eingefügt, dann werden die auf ELSE folgenden Anweisungen ausgeführt. Gibt es keine ELSE-Anweisung, dann werden keine Anweisungen ausgeführt.

Format

```
IF  BooleanExpression1  THEN
      StatementList1;
[ELSEIF  BooleanExpression2  THEN   (*optionaler Teil*)
      StatementList2;]
[ELSE                         (*optionaler Teil*)
      StatementList3;]
END_IF;
```

mit:

BooleanExpression   Ein beliebiger Ausdruck, der einen Booleschen Wert ergibt.

StatementList   Eine beliebige Folge von strukturierten Textanweisungen.

Arbeitsweise

Die nachstehende Bearbeitungsfolge ergibt sich, wenn beide optionalen Teile vorhanden sind:

- Ist BooleanExpression1 TRUE, dann wird StatementList1 ausgeführt. Die Programmausführung wird fortgesetzt mit der Anweisung, die auf das Schlüsselwort END_IF folgt.
- Ist BooleanExpression1 FALSE und BooleanExpression2 TRUE, dann wird StatmentList2 ausgeführt. Die Programmausführung wird fortgesetzt mit der Anweisung, die auf das Schlüsselwort END_IF folgt.
- Sind beide Booleschen Ausdrücke FALSE, dann wird StatmentList3 ausgeführt. Die Programmausführung wird fortgesetzt mit der Anweisung, die auf das Schlüsselwort END_IF folgt.

Gibt es keinen optionalen Teil, dann wird die Programmausführung mit der Anweisung fortgesetzt, die auf das Schlüsselwort END_IF folgt.

Beispiel

Abhängig von dem Wert des E/A-Punktes schreibt das nachstehende Codeteil Text in die Stringvariable "Message".

```
IF Input01 < 10.0 THEN
      Message := 'Warnung unterer Grenzwert';
ELSEIF Input02 > 90.0 THEN
      Message := 'Warnung oberer Grenzwert';
ELSE
      Message := 'Grenzwerte OK';
END_IF;
```

INCLUDE

Beim Syntaxcheck der strukturierten Textdatei nimmt die Anweisung INCLUDE Anweisungen von einer externen Datei auf. Die externe Datei kann strukturierten Text oder Anweisungslisten-Anweisungen enthalten.

Format

```
INCLUDE  FullFilePath;
```

mit:

FullFilePath ist eine Zeichenfolge, die Pfad und Namen der externen Datei angibt.

Beispiel

```
INCLUDE 'C:\ST_Files\ST_File1.TXT';
```

FOR-Anweisung

Die FOR-Schleife führt wiederholt (iterativ) eine im FOR…END_FOR Konstrukt enthaltene Anweisungsliste aus. Es ist hilfreich, wenn die Anzahl Iterationsschritte vorhersehbar ist (z.B. zur Initialisierung eines Felds). Die Anzahl Iterationsschritte beruht auf dem Wert einer Laufvariablen, die von der FOR -Anweisung von einem Anfangswert aus bis zu einem Endwert erhöht oder erniedrigt wird. Sie können als Laufvariable auch einen Ausdruck verwenden, dessen Wert jedoch 0 oder größer sein muß. In der Standardeinstellung erhöht jeder Iterationsschritt der FOR-Anweisung den Wert der Laufvariablen um 1. Über einen optionalen BY-Teil des Konstrukts kann über eine von Null verschiedene ganze Zahl oder über eine Ausdruck, dessen Ergebnis eine positive oder negative ganze Zahl ist, ein Wert angegeben werden, um den die Laufvariable erhöht oder erniedrigt wird.

Vor jedem Iterationsschritt überprüft die FOR-Anweisung die Laufvariable. Die Anweisungen innerhalb des FOR…END_FOR Konstrukts werden nur ausgeführt,

solange der aktuelle Wert der Laufvariablen den angegebenen Endwert nicht überschritten hat.

Das Schlüsselwort END_FOR veranlaßt das System, am Ende jedes Iterationsschrittes der FOR-Schleife einen E/A-Zyklus durchzuführen. Alternativ können Sie zum Durchfahren einer Schleife ohne E/A-Zyklus das Schlüsselwort END_FOR_NOWAIT verwenden.

Format

```
FOR Int_Variable := Ausdruck TO Ausdruck [BY Ausdruck] DO
        Anweisungsliste;
END_FOR;
```

mit:

Int_Variable Eine ganzzahlige Variable.

Ausdruck Ein Einzelwert, Ausdruck, oder komplexer Ausdruck vom gleichen Datentyp wie Int_Variable.

Anweisungsliste Eine beliebige Liste von strukturierten Textanweisungen.

Beispiele

- Der nachstehende Codeteil initialisiert ein Feld (mit 100 Elementen), indem es einen Wert 10 in alle Feldelemente einträgt. Da diese Operation von der E/A unabhängig ist, wird das Schlüsselwort END_FOR_NOWAIT verwendet.

```
FOR index := 1 TO 100
        Array01[index] := 10;
END_FOR_NOWAIT;
```

- Der nachstehende Codeteil weist über zehn E/A-Zyklen Feldelementen die Werte eines E/A-Punkten zu. Der letzte Eintrag erfolgt in das Feldelement mit dem kleinsten Index. Da nach jedem Schleifendurchlauf ein E/A-Zyklus durchgeführt werden soll, wird das Schlüsselwort END_FOR verwendet.

```
FOR index := 10 TO 1 BY -1 DO
        ArrayInput[index] := Input01;
END_FOR;
```

Funktionsaufruf

Der Funktionsaufruf im strukturierten Text führt einen vordefinierten Algorithmus aus, der eine arithmetische Operation, eine Operation mit einer Zeichenfolge oder Bitfolge, oder eine sonstige Operation durchführt. Der Funktionsaufruf besteht aus dem Namen der Funktion oder des Funktionsblocks, gefolgt von den notwendigen Ein- und Ausgangsparametern, die in Klammern eingeschlossen sind.

Funktionsformat

```
FunctionName(Parameter1, Parameter2, . . .); (*unbenannte
Parameter*)
      oder
FunctionName(P1:=Parameter1, Parameter1, . . .);
(*benannte Parameter*)
```

Funktionsblockformat

```
FunctionBlockName(P1:=Parameter1, P2:=Parameter1, . . .);
```

Beispiel 1

Dieser Codeteil zeigt den TAN-Funktionsaufruf.

```
Result := TAN( AnyReal );
```

Beispiel 2

Dieser Codeteil zeigt eine Funktion mit benannten Parametern.

```
StringB := LEFT(IN:= StringA, L:=  VarI);
```

LABEL

Diese Anweisung wird nur innerhalb von Ablaufketten-Schritten verwendet. Besitzt eine Aktionsmarke eine zugehörige LABEL-Anweisung im Ablaufketten-Schritt, dann läuft diese Aktion erst, wenn die Marke aufgetreten ist.

Format

ActionLabel:: Anweisung;

mit:

ActionLabel	Der Name einer Aktionsmarke.
Anweisung	Eine beliebige Anweisung in strukturiertem Text.

Beispiel

Der nachstehende Codeteil veranlaßt, daß eine Aktion mit der Marke ActionLabel abläuft.

ActionLabel:: VarString1:= "Aktion sollte ablaufen";

REPEAT-Anweisung

Die REPEAT-Schleife führt wiederholt (iterativ) eine im REPEAT…END_REPEAT Konstrukt enthaltene Anweisungsliste solange aus, bis eine Ausgangsbedingung erfüllt ist. Sie führt zuerst die Anweisungsliste aus und prüft dann erst die Ausgangsbedingung ab. Dieser Schleifenkonstrukt ist hilfreich, wenn die Anweisungsliste mindestens einmal ausgeführt werden muß.

Format

```
REPEAT
      StatementList;
UNTIL  BooleanExpression  END_REPEAT;
```

mit:

BooleanExpression Ein beliebiger Ausdruck, der einen Booleschen Wert ergibt.

StatementList Eine beliebige Folge von strukturierten Textanweisungen.

Arbeitsweise

Die Anweisungsliste StatementList wird ausgeführt. Ist BooleanExpression FALSE, dann wird die Schleife wiederholt. Ist BooleanExpression dagegen TRUE, wird die Schleife verlassen. Die Anweisungsliste wird mindestens einmal ausgeführt, da BooleanExpression erst am Ende der Schleife ausgewertet wird.

Das Schlüsselwort END_REPEAT veranlaßt das System, am Ende jedes Iterationsschrittes der REPEAT-Schleife einen E/A-Zyklus durchzuführen. Alternativ können Sie zum Durchfahren einer Schleife ohne E/A-Zyklus das Schlüsselwort END_REPEAT_NOWAIT verwenden.

Hinweis

Insbesondere, wenn Sie das Schlüsselwort END_REPEAT_NOWAIT verwenden, können Sie eine Endlosschleife erzeugen, die (da das Steuerungssystem mit höchster Priorität abläuft) die Kontrolle nicht mehr zurück an das Betriebssystem gibt. Um Endlosschleifen zu vermeiden müssen Sie sicherstellen, daß der Boolesche Ausdruck eine bestimmte Ausgangsbedingung liefert.

Beispiel

Der nachstehende Codeteil liest solange Werte aus einem Feld aus, bis ein Wert größer als 5.0 angetroffen wird (oder die obere Feldgrenze erreicht wird). Da mindestens ein Feldwert gelesen werden muß, wird die REPEAT Schleife verwendet.

```
REPEAT
      Value:=Array01[Index];
      Index:=Index+1;
UNTIL  Value > 5.0  OR Index >= UpperBound
END_REPEAT_NOWAIT;
```

SCAN

Die SCAN-Anweisung unterbricht die Programmausführung solange ein E/A-Zyklus stattfindet.

Format

```
SCAN;
```

Beispiel

```
Anweisung1;
SCAN;       (*Anweisungen 2 & 3 werden erst nach einem
E/A-Zyklus ausgeführt.*)
Anweisung2;
Anweisung3;
```

WHILE-Anweisung

Die WHILE-Schleife führt wiederholt (iterativ) eine im Konstrukt WHILE…END_WHILE enthaltene Anweisungsliste solange aus, bis eine angegebene Bedingung TRUE ist. Zuerst wird die Bedingung geprüft, dann die Anweisungsliste entsprechend dem Zustand der Bedingung ausgeführt. Dieser Schleifenkonstrukt ist hilfreich, wenn die Anweisungsliste nicht unbedingt ausgeführt werden muß.

Format

```
WHILE  BooleanExpression DO
      StatementList;
END_WHILE;
```

mit:

BooleanExpression Ein beliebiger Ausdruck, der einen Booleschen Wert ergibt.

StatementList Eine beliebige Folge von strukturierten Textanweisungen.

Arbeitsweise

Ist BooleanExpression FALSE, dann wird die Schleife sofort verlassen. Ist BooleanExpression TRUE, dann wird StatementList ausgeführt und die Schleife wiederholt. Es kann vorkommen, daß die Anweisungsliste überhaupt nicht ausgeführt wird, da BooleanExpression am Anfang der Schleife ausgewertet wird.

Die Anweisung END_WHILE veranlaßt das System, am Ende jedes Durchlaufs der WHILE-Schleife einen E/A-Zyklus durchzuführen. Alternativ können Sie zum Durchfahren einer Schleife ohne E/A-Zyklus die Anweisung END_WHILE_NOWAIT verwenden.

Hinweis

Insbesondere, wenn Sie das Schlüsselwort END_WHILE_NOWAIT verwenden, können Sie eine Endlosschleife erzeugen, die (da das Steuerungssystem mit höchster Priorität abläuft) die Kontrolle nicht mehr zurück an das Betriebssystem gibt. Um Endlosschleifen zu vermeiden müssen Sie sicherstellen, daß der Boolesche Ausdruck eine bestimmte Ausgangsbedingung liefert.

Beispiel

Der nachstehende Codeteil hält einen Ausgangspunkt solange wie der Eingangspunkt EndLimit FALSE bleibt. Es wird eine WHILE-Schleife verwendet, da es keinen Grund gibt, die Anweisungsliste zu bearbeiten, wenn EndLimit = TRUE. Die END_WHILE-Schleife wird verwendet, da die E/A-Punkte abgefragt werden müssen.

```
WHILE NOT EndLimit DO
      MoveForward := TRUE;
END_WHILE;
MoveForward := FALSE;
```

Strukturierte Textoperatoren

Die nachstehende Tabelle führt die Operatoren auf, die Sie in einem Ausdruck verwenden können. Die Priorität legt fest, in welcher Reihenfolge die Operatoren im Ausdruck bearbeitet werden. Der Operator mit der höchsten Priorität wird zuerst angewandt, anschließend der Operator mit der zweithöchsten Priorität. Operatoren mit gleicher Priorität werden von links nach rechts ausgewertet.

Operator	Symbol	Priorität
Runde Klammer	( )	1
Funktionsauswertung	Bezeichner (Argumentliste) z.B. LN (A), ABS (X)	2
Potenzierung	**, POW	3
Negierung	-	4
Komplement	NOT	4
Multiplikation	*	5
Division	/	5
Modulo	MOD	5
Addition	+	6
Subtraktion	-	6
Vergleich	<, >, <=, >=	7
Gleichheit	=	8
Ungleichheit	<>	8
Booleschesbitweises UND	AND	9
Boolesches/bitweises Exklusiv-ODER	XOR	10
Boolesches/bitweises ODER	OR	11

Diese Symbole besitzen folgende Funktionen:

- := weist einen Ausdruck einem Symbol zu
- ; wird benötigt, um das Ende einer Anweisung zu bezeichnen
- [ ] wird zur Feldindizierung benutzt, wenn der Feldindex eine ganze Zahl ist. Der nachstehende Ausdruck setzt zum Beispiel das erste Element eines Feldes auf den Wert j+10: intarray[i] = j + 10;
- :: wird zur Kennzeichnung einer Marke verwendet. Zum Beispiel: spray_on:: Nach einer Marke muß in der gleichen Zeile eine Anweisung stehen.
- (* *) kennzeichnet einen Kommentar. Zum Beispiel: (*dies ist ein Kommentar.*)

Anweisungstypen

Die strukturierten Textanweisungen, die für die eigentliche Programmausführung sorgen, bestehen aus folgenden Typen:

- **Zuweisung -** Setzt ein Objekt auf einen angegebenen Wert.
- **CASE -** Sorgt für die bedingte Ausführung einer Reihe von Anweisungen.
- **Kommentar -** Einbeziehung von Kommentaren in das Programm.
- **EXIT -** Beendet die Iterationen, ehe die Abschlußbedingung TRUE wird.
- **FOR -** Zeigt an, daß eine Anweisungsfolge wiederholt entsprechend dem Wert eines Steuersymbols ausgeführt wird.
- **Funktionsaufruf -** Ruft eine Funktion zur Ausführung auf.
- **IF -** Gibt an, daß eine oder mehrere Anweisungen bedingt ausgeführt werden sollen.
- **INCLUDE -** Führt eine Reihe von Anweisungen aus einer externen Datei aus.
- **REPEAT -** Zeigt an, daß eine Anweisungsfolge wiederholt solange ausgeführt werden soll, bis ein Boolescher Ausdruck TRUE wird.
- **SCAN -** Stoppt die Ausführung von strukturiertem Text solange, bis ein E/A-Zyklus durchgeführt wurde.
- **WHILE -** Zeigt an, daß eine Anweisungsfolge wiederholt solange ausgeführt werden soll, bis ein Boolescher Ausdruck FALSE wird.

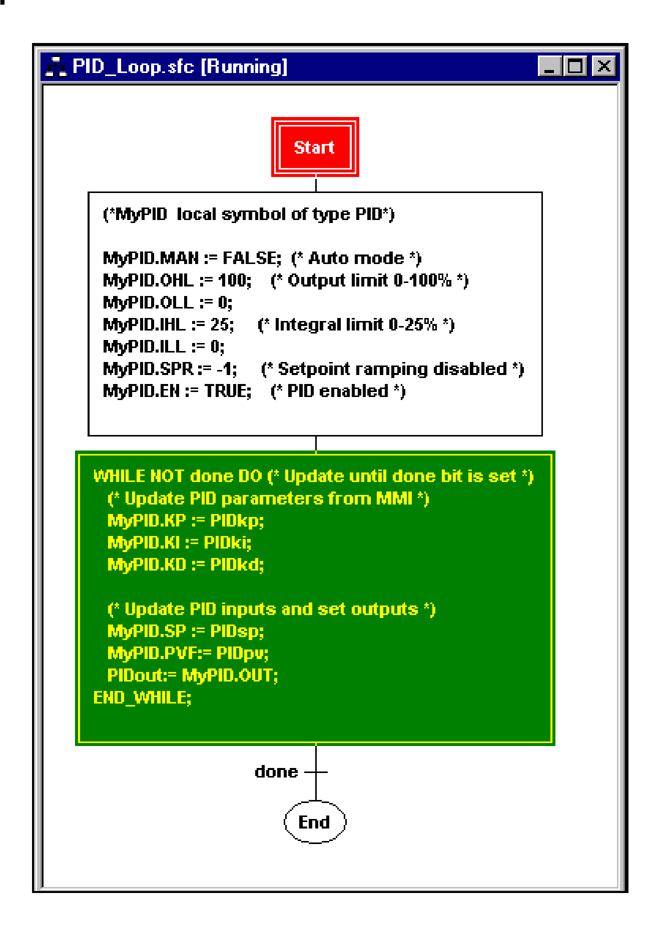
Einsatz von PIDs in einem Anwenderprogramm

Verwendung von PID-Elementen

Ein PID-Element ist eine Anweisung für den automatischen Regelbetrieb von kontinuierlichen Prozeß-Regelkreisen. Für jeden Kreis führt die Anweisung Proportionalregelung und wahlweise Integralregelung, Differentialregelung, oder beide durch:

- Proportionalregelung - bewirkt die Veränderung eines Ausgangssignals im direkten Verhältnis der Regeldifferenzänderung.
- Integralregelung - bewirkt die Veränderung eines Ausgangssignals als Funktion des Integrals der Regeldifferenz über die Zeitdauer.
- Differentialregelung - bewirkt die Veränderung eines Ausgangssignals als Funktion der Änderungsgeschwindigkeit der Regeldifferenz.

PID-Beispiel



Hinweise zur Verwendung von PIDs

1. Ein Symbol vom Typ PID sollte nur als lokale Variable definiert werden. Ein globales PID-Symbol wird bei laufendem Programm einmal pro E/A-Zyklus ausgewertet. Dies erzeugt instabile und unvorhersehbare Ergebnisse an den PID-Ausgängen.

2. Ist SPR < 0, tritt keine lineare Sollwertänderung auf und der PID-Kreis verwendet den Rohwert des Sollwertes.

3. Ist MAN = FALSE (AUTO-Modus), wird bei jeder Auswertung der PID-Funktion OVR = OUT gesetzt.

4. IHL, ILL und IHLD beeinträchtigen die stoßfreie Ergebnisübertragung.

5. FF kann als statische bleibende Regelabweichung oder als dynamische Störgrößenaufschaltung oder als beides verwendet werden.

6. Verwendung von PID in strukturiertem Text :

   A. Erzeugen Sie eine lokale Variable vom Typ PID.

   B. Stellen Sie die Anfangswerte von KP, KI, KD, OHL, OLL, IHL, ILL, IHLD, SPR, MAN, FF ein.

   C. Erstellen Sie ein Programm zur dauernden Aktualisierung von SP, PVF und FF (optional) und zum Speichern von OUT.

   D. Setzen Sie EN , um die Aktualisierung der PID-Funktion zu starten.

Steuerung des Ablaufs von Anwenderprogrammen in Kontaktplan und strukturiertem Text

Über den Programmsteuerungsblock (PRGCB) kann ein AS-Anwenderprogramm andere KOP- und AS-Anwenderprogramme kompilieren und deren Ausführung steuern.

Verwendung von PRGCB

1. Erstellen Sie eine Ablaufkette und definieren Sie eine lokale Variable (z.B. prgcb) vom Typ PRGCB.

2. Speichern Sie eine Zeichenfolge (STRING), die Pfad und Dateinamen des Programms enthält, das in prgcb.NAME gesteuert werden soll.

3. Verwenden Sie die Booleschen Eingänge von PRGCB, um das Programm zu steuern.

4. Verwenden Sie die Ausgänge von PRGCB, um Informationen über das Programm zu erhalten.

Verwendung des PRGCB Statuscodes

Mit PRGCB.STATUS können Sie die verschiedenen Zustände der Steuerungsprogramm-Ausführung überwachen. Diese eindeutige Variable gibt einen ganzzahligen Wert zurück, der sowohl aus dem Steuerungsprogramm selbst heraus verwendet als auch als Anzeige am Bildschirm der Bedienerschnittstelle dargestellt werden kann. PRGCB.STATUS liefert folgende Anzeigen:

Wert	Definition	Wert	Definition
0	Initialisiert	8	Programm abgeschlossen
1	Datei nicht gefunden	9	Programm wird abgebrochen
2	Datei offen	10	Programm wurde abgebrochen
3	Datei-Parsing	11	Programm fehlerhaft
4	Parserfehler aufgetreten	12	Programm am Haltepunkt
5	Syntaxanalyse beendet	13	Programm angehalten
6	Programm läuft	14	Programm zurückgespult (REWIND-Funktionalität)
7	Programm gestoppt	15	Programm nicht zurückgespult (REWIND-Funktionalität)

Einsatz der PRGCB-Rückspulfunktion

Nachdem ein Steuerungsprogramm bis zum Ende durchgelaufen ist, läuft es erst wieder, nachdem Sie die lokale Programmsteuerungsvariable prgcb.REWIND gesetzt und dadurch das Programm effektiv zurückgespult haben. Der Rückspulzustand eines Programms kann über die Variable prgcb.STATUS überwacht werden.

PRGCB-Beispiel

Der nachstehende AS-Code startet das Demonstrationsprogramm "Cookie" und startet es am Ende jedesmal wieder neu. Dieses Beispiel verwendet die Eingänge NAME, REWIND und RUN sowie den Ausgang INCYCLE. Andere Boolesche Eingänge und Ausgänge werden in der Online-Hilfe zu PRGCB erläutert.

```
cookie.NAME := "c:\cimplicity\pccontrol\cookie\cookie.sfc";

cookie.REWIND := TRUE;

cookie.RUN := TRUE;

WHILE TRUE DO

    (* Neustart wenn das Programm Cookie anhält *)

    IF NOT cookie.INCYCLE THEN

          cookie.REWIND := TRUE;

           cookie.RUN := TRUE;

    END_IF;


    (*SCAN;*)

    buzzer := FALSE;

    switch1 := TRUE;

END_WHILE;
```

Anwenderprogramme und Symbole überwachen und testen

Syntax eines Programms prüfen

Um die Syntax eines Programms prüfen zu können, muß das Windows NT Betriebssystem aktiv sein und es muß die PC Control Software installiert sein.

Syntaxanalyse bei der aktiven Datei durchführen:

Klicken Sie auf *Ausführen* und wählen die Option *Syntaxanalyse*.

Werden Syntaxfehler festgestellt, erscheint eine Meldung und der Fehlerort wird invers dargestellt. Ist der Syntaxcheck des Programms abgeschlossen, wird der Syntaxstatus im Fenstertitel der aktiven Datei aktualisiert.

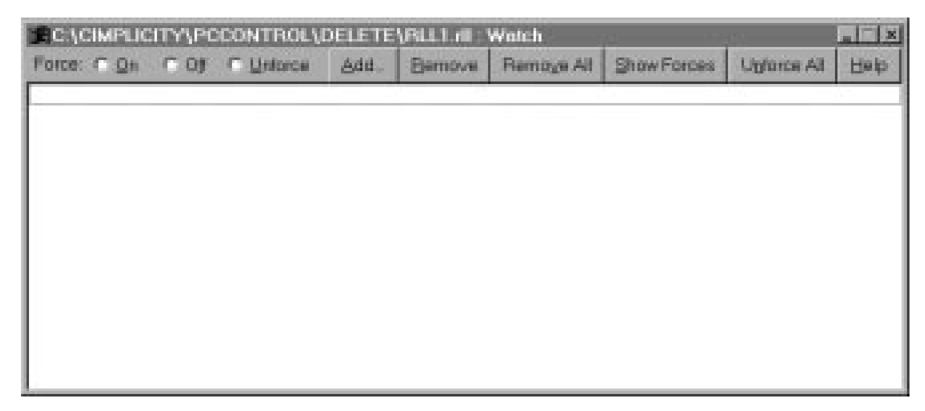
Symbole beobachten und fixieren

Das in den nachstehenden Abbildungen gezeigte Überwachungsfenster zeigt lokale und globale Symbole und deren Status zur Laufzeit an.

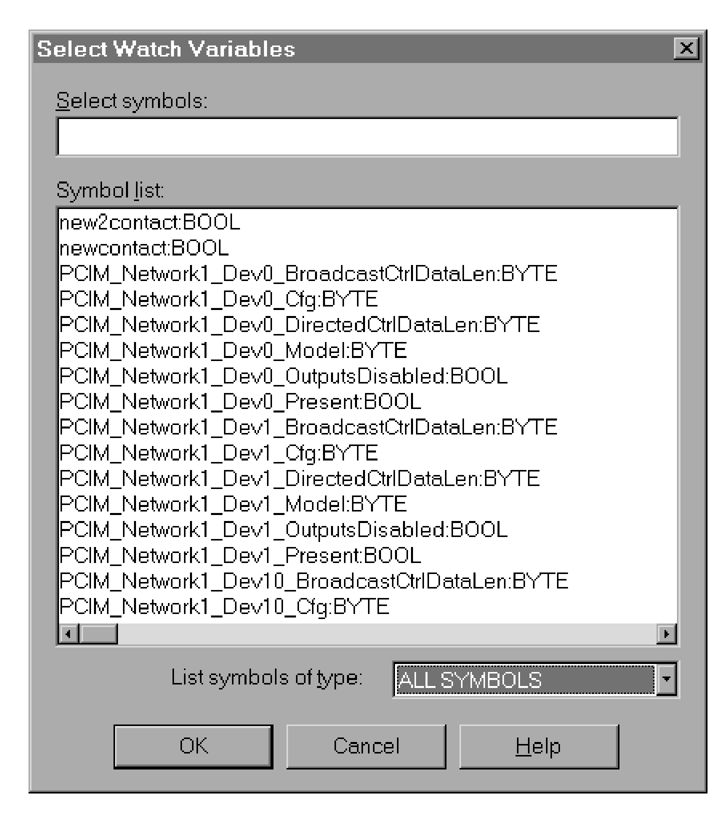
Überwachungsfenster anzeigen oder ausblenden:

Klicken Sie auf *Anzeigen* und wählen dann *Variablen beobachten/fixieren* oder klicken Sie auf das Überwachungsfenstersymbol.





Klicken Sie auf die Schaltfläche *Hinzufügen*, um Variablen zur Anzeige im Überwachungsfenster auszuwählen. Hierauf erscheint die Dialogbox "Beobachtungsvariablen auswählen".



Wählen Sie die Variablen, die Sie überwachen wollen, und klicken dann auf OK. Die ausgewählten Variablen werden im Überwachungsfenster angezeigt.

Lauf mit Austesten

Die Programmoption "Lauf mit Austesten" gestattet es, ein Programm zusammenhängend solange ablaufen zu lassen, bis eine BREAK-Anweisung angetroffen wird. Das Programm wird angehalten, wenn die BREAK-Anweisung angetroffen wird. Ab diesem Punkt können Sie dann das Programm abbrechen oder im Einzelschrittbetrieb fortsetzen.

Einzelschrittbearbeitung eines Programms

Mit der Einzelschritt-Programmoption können Sie bei der Programmausführung jeweils einen vollständigen Zyklus ablaufen lassen. Vor der Einzelschrittbearbeitung muß das Programm angehalten werden. Die Betriebsarten der übrigen Programme im Projekt sind irrelevant.

Einzelschrittbearbeitung eines Programms:

1. Wählen Sie das Programm im Programmeditor aus. Klicken Sie bei laufendem Programm in der Menüleiste auf *Ausführen* und wählen Sie dann *Stop*.
2. Öffnen Sie für das Programm ein Überwachungsfenster und fügen Sie alle Symbole hinzu, die Sie beim Ablaufen des Programms überwachen wollen. Diese Einstellungen sind optional.
3. Klicken Sie in der Menüleiste auf *Ausführen* und wählen dann *Einzelschritt*. Das Programm geht für einen Zyklus in RUN-Modus, dann weiter in den Unterbrechungsmodus. Die gesamte vom Programm kontrollierte E/A wird während des Programmzyklus aktualisiert.

Fehlermodus und Fehlerzustände löschen

Zum Löschen von Notaus und anderen E/A-Fehlern wählen Sie *Notaus rücksetzen* aus dem Menü *Ausführen*.

Abschnitt 5: Anweisungslisten-Programmierung

Übersicht

Anweisungsliste (AWL) ist eine textstrukturierte Programmiersprache entsprechend IEC 1131-3. Das Format der Anweisungsliste ist ähnlich dem Format einer Assemblersprache. Mit dem Anweisungslisteneditor können Sie eigenständige Anweisungslistenprogramme erstellen. Den Anweisungslisteneditor können Sie aus dem Programmeditor heraus aufrufen. Er besitzt typische Texteditierfunktionen wie Ausschneiden, Kopieren, Einfügen, suchen und Ersetzen. Darüberhinaus besitzt er Werkzeuge und Befehle zum automatischen Einfügen von Anweisungslisten-Befehlen und -Funktionen.

Dieser Abschnitt erläutert, wie Sie mit dem Anweisungslisteneditor Anweisungslistenprogramme erstellen können. Es wird angenommen, daß Sie die allgemeine Arbeitsweise eines Programmeditors kennen und mit Anweisungslisten etwas vertraut sind.

Hinweis Anweisungslistenprogramm sind kontinuierlich ablaufende Programme (einmal pro Zyklus).

Öffnen eines Anweisungslistendokuments

Zum Öffnen eines bestehendes Dokuments wählen Sie *Editor öffnen* im Programmeditormenü *Datei* und selektieren das Dokument mit der daraufhin erscheinenden Dialogbox *Öffnen*.

Zum Öffnen eines neuen Dokuments wählen Sie *Editor Neu* im Programmeditormenü *Datei* und wählen *Anweisungslisten-Dokument* aus der daraufhin erscheinenden Dialogbox *Neu*.

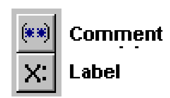
Anweisungen eingeben

Eingabe von Hand

Anweisungen können eingegeben werden, indem Operator oder Funktionsaufruf und die zugehörigen Operanden oder Parameter über die Tastatur eingegeben werden. Beachten Sie bei Syntax, Operatoren und Modifizierer die Angaben in **Sprachübersicht** und die Funktionsaufrufsyntax in der **Sprachreferenz**.

Zubehörleiste

Die Zubehörleiste zeigt Befehle in graphischer Form an. Sie bietet eine alternative Methode der Befehlseingabe. Die Zubehörleiste erscheint nur, wenn Sie über *Zubehörleiste* im Menü *Anzeige* freigegeben wurde. Die nachstehende Abbidlung zeigt die Funktionen der Zubehörleiste. Die frei bewegliche Zubehörleiste kann losgelöst und nach Bedarf plaziert werden.



Menü zum Einfügen von AWL-Anweisungen

Bei aktivem Anweisungslisteneditor enthält das Programmeditormenü *Bearbeiten* ein Element *AWL-Anweisung einfügen*, das Anweisungslisten-Anweisungen (Operatoren) auflistet. Wählen Sie die benötigte Anweisung und sie wird automatisch an der Cursorposition eingefügt.

Menü zum Einfügen von AWL-Funktionen

Bei aktivem Anweisungslisteneditor enthält das Programmeditormenü *Bearbeiten* ein Element *AWL-Funktionen einfügen*, das die Standardfunktionen aufführt, die ím Zusammenhang mit der Anweisungsliste verwendet werden können. Wählen Sie die benötigte Funktion und sie wird automatisch an der Cursorposition in richtiger Syntax eingefügt. Ersetzen Sie alle Parameter mit denen, die Sie im Symbolmanager definiert haben.

Weitere Information zur Verwendung von Funktionen und Funktionsblöcken in der Anweisungsliste finden Sie in **Sprachübersicht**.

Anweisungen bearbeiten

Der Anweisungslisteneditor unterstützt die allgemeinen Editorfunktionen, wie zum Beispiel Ausschneiden, Kopieren, Einfügen, Suchen und Ersetzen. Diese Befehle finden Sie unter *Bearbeiten* und im Kontextmenü.

Sprachübersicht

Syntax der Anweisungsliste

Ein Anweisungslistenprogramm besteht aus einer Liste mit Anweisungen. Jede Anweisung beginnt auf einer neuen Zeile und kann eine Marke, einen Operator, Operatormodifizierer, Operanden und Kommentarfelder enthalten:

Marke	Operator	Operand	Kommentar
Beispiel:	LD	Sym01	(*Lade Wert Sym01 in Akkumulator*)
	ADD	Sym02	(*Addiere Sym02 dazu*)
			(*Die Summe ist jetzt das aktuelle
Ergebnis*)			

Kommentare können nur am Ende einer Zeile stehen. Zwischen den Anweisungszeilen sind Leerzeilen erlaubt. Das aktuelle Ergebnis wird in einem Akkumulator verwaltet. Anweisungen arbeiten wie folgt:

aktuelles_ergebnis:= aktuelles_ergebnis OPERATOR Operand

wobei aktuelles_ergebnis immer links vom Operator steht.

Operatoren

Die nachstehende Tabelle listet die in der Anweisungslisten-Sprache verwendeten Operatoren auf:

Operator	Modifizierer	Operand	Beschreibung
LD	N		Setzt aktuelles Ergebnis gleich dem Operanden.
ST	N		Speichert das aktuelle Ergebnis in der Operandenadresse.
S	Hinweis 1	BOOL	Setzt Booleschen Operanden auf 1.
R	Hinweis 1	BOOL	Setzt Booleschen Operanden zurück auf 0.
AND	N, (	BOOL	Boolesches UND.
OR	N, (	BOOL	Boolesches ODER.
XOR	N, (	BOOL	Boolesches Exklusiv-ODER.
ADD	(		Addition.
SUB	(		Subtraktion.
MUL	(		Multiplikation.
DIV	(		Division.
GT	(		Vergleich "größer als" (>).
GE	(		Vergleich "größer als oder gleich" (>=).
EQ	(		Vergleich "gleich" (=).
NE	(		Vergleich "ungleich" (<>).
LE	(		Vergleich "kleiner als" (<).
LT	(		Vergleich "kleiner als oder gleich" (<=).
JMP	C, N	LABEL	Sprung zu Marke.
CAL	C, N	NAME	Funktions- und Funktionsblock-Aufruf.
)			Auswertung zeitversetzter Operation.

Hinweis

1. Wird nur ausgeführt, wenn das aktuelle Ergebnis eine Boolesche "1" ist.

Tabellenherkunft: IEC 1131-3 (Teil 3 der IEC-Norm 1131 für speicherprogrammierbare Steuerungen), International Electrotechnical Commission.

Modifizierer

Operatoren können folgende Modifizierer annehmen:

N Boolesche Negierung des Operanden. Zum Beispiel:

```
ORN Bool1
```

ist äquivalent zu ergebnis:= ergebnis OR NOT Bool1.

C Diese Anweisung wird nur ausgeführt, wenn der Wert von aktuelles_ergebnis eine Boolesche "1" ist. (Oder eine Boolesche "0", wenn der Modifizierer N ebenfalls benutzt wird. Zum Beispiel:

```
LD      Wert
JMPC    Sort
```

ist äquivalent zu "IF Wert ist TRUE, springe zu Sort, ELSE weiter mit Bearbeitung."

(Verzögert die Auswertung des Operators bis zum Antreffen eines rechten Klammeroperators ")". Zum Beispiel:

```
MUL(    Num1
ADD     Num2
)
```

ist äquivalent zu
ergebnis:= ergebnis * (Num1 + Num2).

Funktionen und Funktionsblöcke

Funktionsaufrufe

Funktionsaufrufe können in den folgenden Formen erfolgen:

CALC ruft eine Funktion auf, wenn das aktuelle Ergebnis (Akkumulatorwert) TRUE (1) ist.

CAL ruft die Funktion immer auf..

CALCN ruft die Funktion nur auf, wenn das aktuelle Ergebnis (Akkumulatorwert) FALSE (0) ist.

Die Funktionsaufrufsyntax mit CALC wird nachstehend dargestellt:

Beispiel:

```
CALC   ROL(OUT:= VarBit, IN:= BitString, N:= RotateNum)
```

FunktionBeispiel2:

```
CALC   ADD(OUT:= VarNum, Num1, Num2)
```

Funktionsblock

Funktionsblöcke werden mit der folgenden Aufrufform aufgerufen:

CAL ruft den Funktionsblock immer auf.

Die folgenden Beispiele zeigen die für einen Standard-Funktionsblock verwendete Syntax, wobei ctu1 und ton1 benannte Instancen der Funktionsblöcke CTU und TON sind.

LD 10 ST ctu1.PV LD TRUE ST ctu1.EN LD In3 ST ctu1.CU CAL ctu1 LD ctu1.Q ST Out3	LD TRUE ST ton1.IN ST ton1.EN LD t#30s ST ton1.PT CAL ton1 LD ton1.Q ST Out2

Die Beschreibungen und Parameter (PV, EN, CU, Q, usw.) der Standard-Funktionsblöcke finden Sie unter **Sprachreferenz**.

Akkumulatorbeziehungen

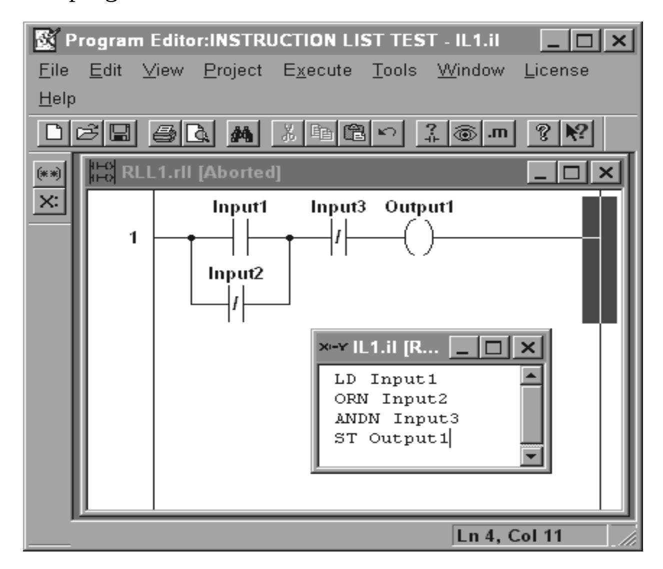
Bei Funktionsblöcken gibt es keine Beziehung zwischen Akkumulator und Funktionsblock. Der Funktionsblock wird immer aufgerufen und der Akkumulator wird nicht in den Funktionsblock hineingeschoben, Nach der Rückkehr aus dem Funktionsblock hat der Akkumulator noch den gleichen Wert wie vor dem Aufruf des Funktionsblocks.

Bei Funktionen hat der Akkumulator keine Wirkung auf die Funktionseingänge. Wird CALC verwendet, dann wird die Funktion nicht aufgerufen, wenn der Akkumulator FALSE ist.

Wird CALCN verwendet, dann wird die Funktion nicht aufgerufen, wenn der Akkumulator TRUE ist. Ist bei Rückkehr aus der Funktion der Rückgabewert ein BOOLESCHE WERT, dann wird der Rückgabewert automatisch in den Akkumulator geladen. Ist der Rückgabewert der Funktion kein Boolescher Wert, kann der Rückgabewert mit der Funktionssyntax func1 (OUT:=outvar1) in eine Variable gespeichert werden. In diesem Fall wird der Rückgabewert nur dann in outvar1 gespeichert, wenn die Funktion aktuell aufgerufen ist (CAL, CALC mit Akkumulator TRUE, oder CALCN mit Akkumulator FALSE). Ist der Rückgabewert der Funktion kein Boolescher Wert, dann sollte der Akkumulator automatisch mit dem invertierten Wert des Booleschen Systemfehlersymbols (RTERROR) geladen werden. Wird die Funktion aufgerufen und meldet einen Fehler, wird der Akkumulator nach Rückkehr aus der Funktion mit einem Wert FALSE geladen. Wird die Funktion aufgerufen und meldet keinen Fehler, wird der Akkumulator mit einem Wert TRUE geladen.

Programmbeispiele

Die nachstehende Abbildung zeigt äquivalente Kontaktplan- und Anweisungslistenprogramme.



Beispiele zu den verwendeten Anweisungslisten-Operatoren finden Sie in der Online-Hilfe für PC Control.

Allgemeine Operationsbeispiele

```
LD  Format  (* 0: KONVERTIERE F TO C,  1: KONVERTIERE C TO F *)
JMPC        CtoF            (* SPRINGE ZU ENTSPRECHENDER FUNKTION  *)


LD  .555            (* LADE MULTIPLIZIERER  *)
MUL(        Fahrenheit     (* ERGIBT: (Fahrenheit-32)*.555)  *)
SUB 32
)
ST  Celsius         (* SPEICHERE ERGEBNIS IN Celsius  *)
JMP Done                    (* OPERATION BEENDET, GEHE ZU DATEIENDE
*)
CtoF:       LD      32              (* LADE OPERAND  *)
ADD(        Celsius         (* ERGIBT: (Celsius * 1.8)+32  *)
MUL 1.8
)
ST  Fahrenheit              (* SPEICHERE ERGEBNIS IN Fahrenheit  *)


Done:       LD      1               (* KONVERTIERUNG BEENDET  *)


LD  Run_Timer               (* LADE TRUE IN DAS REGISTER  *)
ANDN        FALSE           (* ERGIBT: TRUE AND NOT FALSE  *)
ST  Enable          (* SPEICHERE IN Enable  *)
```

Beispiel eines Funktionsblockaufrufs

```
LD  Enable        (* LADE Enable IN DAS REGISTER  *)
ST  ton1.IN       (* SPEICHERE REGISTERWERT IN ton1.IN  *)
ST  ton1.EN       (* SPEICHERE REGISTERWERT IN ton1.EN  *)
LD  t#30s         (* LADE t#30s IN DAS REGISTER  *)
ST  ton1.PT       (* SPEICHERE REGISTERWERT IN ton1.PT  *)
CAL ton1          (* AUFRUF VON ton1, LÄUFT FÜR 30s  *)
LD  ton1.Q        (* LADE BOOLESCHEN AUSGANG IN REGISTER *)
ST  Enable_Out    (* SPEICHERE REGISTERWERT IN Enable_Out  *)
```

Abschnitt 6: Online-Editieren

Online-Editieren - Arbeitsweise

Online-Editieren bezieht sich auf das Durchführen von Änderungen in einem laufenden Programm. Beim Online-Editieren gibt es zwei Betriebsarten: nahtlos und nicht nahtlos. Der eingegebene Online-Editiermodus hängt entsprechend den Ausführungen unter "Regeln” von der Art der durchgeführten Bearbeitung ab.

Beim nahtlosen Online-Editieren können in einem Programm Änderungen durchgeführt werden, wobei diese Änderungen sich übergangslos im laufenden Programm widerspiegeln (d.h. ohne den Laufzeitbetrieb zu stören). Nehmen wir an, daß ein Programm läuft und Sie dieses Programm abändern: Kann diese Änderung nahtlos online bearbeitet werden, geht der Editor in den nahtlosen Online-Editiermodus. Das Online-Editier-Bedienelement erscheint mit vier aktiven Schaltflächen:

Programm-Neustart Startet das Programm neu und setzt die E/A zurück. Das aktuelle Programm wird abgebrochen und das neue Programm aktiviert; die E/A wird im Prozeß abgeschaltet.

Änderungen aktivieren Die alte Programmversion wird nahtlos durch die neue Version ersetzt. Die E/A-Aktualisierung wird nahtlos fortgesetzt. Bei Bedarf wird zuerst eine Syntaxanalyse der Änderungen durchgeführt. Das Programm wird nur dann syntaxgeprüft, wenn das Datum von AS- oder Kontaktplandatei jünger ist als das des Programms, für den die Syntaxprüfung durchgeführt wurde. Die Datei wird automatisch gespeichert, wenn sie verändert wurde.

Bei der Aktivierung von Änderungen wird die Online-Editierbox angezeigt, wobei alle Schaltflächen inaktiv sind (solange die Syntaxprüfung der Änderungen durchgeführt wird).

War die Aktivierung von Änderungen erfolgreich, dann verschwindet die Online-Editierbox. Ist die Syntaxprüfung der Änderungen nicht erfolgreich, wird eine Parserfehlermeldung angezeigt und die Online-Editier-Schaltflächen werden wieder aktiviert.

Syntaxanalyse der Änderungen Führt eine Syntaxanalyse der Programmänderungen durch, ohne sie laufen zu lassen. War die Syntaxanalyse der Änderungen erfolgreich, werden die Online-Editierbox-Schaltflächen wieder aktiviert. Tritt ein Syntaxfehler auf, wird eine Parserfehlermeldung angezeigt und die Online-Editier-Schaltflächen werden wieder aktiviert.

Bei der Syntaxanalyse der Änderungen wird die Online-Editierbox angezeigt, wobei alle Schaltflächen inaktiv sind (solange die Syntaxprüfung der Änderungen durchgeführt wird).



Änderungen aufheben Gibt die Quelle zurück zum laufenden Programm und fährt mit dem Selektieren des aktiven Programms fort.

Wenn Sie bei einem laufenden Programm eine Änderung durchführen, die nicht nahtlos implementiert werden kann, dann sind bei der Online-Editierbox nur die Schaltflächen *Programm-Neustart* und *Änderungen aufheben* aktiv.

Wenn eine nicht nahtlose Änderung zum ersten Mal implementiert werden soll, werden Sie gefragt. Haben Sie einmal Ihre Zustimmung gegeben, erfolgt keine weitere Benachrichtigung und die Schaltflächen *Aktivierung von Änderungen* und *Syntaxanalyse der Änderungen* werden gesperrt. War bereits zuvor eine nahtlose Änderung durchgeführt worden, erzwingt die nicht nahtlose Änderung einen Neustart des Programms.

Regeln

Werden Änderungen durchgeführt, die nicht nahtlos im laufenden Programm aktualisiert werden können, wird nicht nahtloses Online-Editieren durchgeführt.

Allgemein

Der Editor geht direkt in den nicht nahtlosen Online-Editiermodus (die Online-Editierbox erscheint mit inaktiven Schaltflächen *Aktivierung von Änderungen* und *Syntaxanalyse der Änderungen*), wenn beim erneuten Start des Editors folgende Bedingungen erfüllt sind:

bei einem aktiven Programm sind Änderungen bearbeitet worden,

die Änderungen wurden gespeichert, ohne sie zu aktivieren (nahtlose Änderungen oder nicht nahtlose Änderungen),

und der Editor wurde geschlossen.

Dies bedeutet, daß nahtlose Online-Bearbeitungsvorgänge nicht über Editiersitzungen hinweg gemerkt werden, daß aber die Änderungen selbst durch die Dateidaten (SST-Datei verglichen mit AS-Datei oder RST-Datei verglichen mit KOP-Datei) gekennzeichnet sind.

Symbole

Sie können neue globale Speichervariablen nahtlos hinzufügen. Hierzu fügen Sie vom Symbolmanager neue globale Speichervariablen hinzu und klicken dann auf *Anwenden* - die neuen globale Speichervariable werden in der Runtimemaschine nahtlos aktiv. Bei jedem Löschen oder Verändern von globalen Speichervariablen oder beim Verändern oder Hinzufügen globaler Anwenderstrukturen müssen die Konfiguration neu aktiviert, alle Programme abgebrochen und die E/A abgeschaltet werden. Bei einem Versuch, eine globale Speichervariable zu löschen oder zu verändern, oder eine globale Anwenderstruktur zu verändern erscheint ein Hinweis, daß durch diese Änderung alle Programme abgebrochen werden müssen.

- Globale und lokale Symbole können hinzugefügt werden.
- Das Programm wird neu gestartet wenn Symbole gelöscht oder verändert werden.
- Das Programm wird neu gestartet wenn Anwenderstrukturen verändert werden.

E/A

- Die Konfiguration muß neu aktiviert werden, wenn die E/A-Treiber verändert werden.

Kontaktplanprogramme

- Beim Editieren von Kontaktplanprogrammen sind beliebige Änderungen möglich. Das gesamte Programm wird erneut syntaxgeprüft und das Programm wird nahtlos ausgelagert. Symbole behalten ihren aktuellen Wert. Die E/A-Aktualisierung wird nahtlos fortgesetzt.

Hinweis

Kontakte, die positive oder negative Umschaltvorgänge erkennen, dürfen bei der ersten Auswertung des Elements keinen Umschaltvorgang verursachen. Bei der Online-Bearbeitung bedeutet dies, daß der erste Zyklus nach dem Online-Editieren den Umschaltvorgang nie erkennt (da dies für das bearbeitete Programm der erste Zyklus ist). Dies kann ein Problem verursachen, wenn das Element auf L-Pegel liegt und während des Zyklus der Online-Bearbeitung auf H-Pegel wechselt. Dieser Umschaltvorgang würde nicht erkannt werden.

Ablaufketten

- Beim bearbeiten von Ablaufketten können Sie den Inhalt von Schritten, Aktionen und Transitionen beliebig verändern. Sie können jedoch weder die Struktur der Ablaufkette verändern (dies würde einen Neustart des Programms erzwingen), noch den Namen eines Elements (Schritt, Aktion oder Transition) ändern. Beim nahtlosen Online-Editieren wird die gesamte Ablaufkette (einschließlich aller Makro-Ablaufketten) erneut syntaxgeprüft, die Programmsegmente, die die Schritte, Aktionen und Transitionen darstellen, werden nahtlos ausgelagert, die Symbole behalten ihre aktuellen Werte, und aktive Schritte, Aktionen und Transitionen bleiben aktiv. Die E/A-Aktualisierung wird nahtlos fortgesetzt.
- Bei Ablaufkettenaktion-Bestimmungszeichen kann nur die Dauer verändert werden. Bei allen anderen Änderungen würde das Programm neu gestartet werden.
- Bei Ablaufketten-Schritten, die strukturierten Text enthalten, bleibt ein abgeschlossener Schritt (die Bearbeitung des strukturierten Textes ist beendet) abgeschlossen, wobei aktive Aktionen weiterhin zyklisch bearbeitet werden. Ist der Schritt nicht abgeschlossen (der strukturierte Text wird noch bearbeitet), dann beginnt die Bearbeitung des strukturierten Textes wieder von vorne.

Dateioperationen

Bei Online-Editen und Dateioperation gibt es einige grundlegende Unverträglichkeiten. Die Art des Online-Editierens stellt sicher, daß alle Variablen, einschließlich den Dateisteuerungsblöcken, ihren aktuellen Zustand während des Online-Editierens beibehalten. Während des Online-Editierens wird der strukturierte Text innerhalb eines Schrittes neu gestartet. Werden während des Online-Editierens ein Satz von strukturierten Textdateibefehlen ausgeführt, dann werden zwar die Befehle neu gestartet, die Datei wird aber nicht geschlossen und auf den Start rückgesetzt. Die Dateioperationen werden daher nach dem Online-Editieren fehlschlagen oder falsch ablaufen.

Beispiel 1

```
    STEP1:
FILE_OPEN (fcb, "test.dat");
WHILE (NOT fcb.EOF) DO
    FILE_READ (fcb, struct1);
END_WHILE;
```

Wird während der WHILE-Schleife ein Online-Editieren durchgeführt, , dann wird STEP1 rückgesetzt und der Befehl FILE_OPEN wird nicht ausgeführt, da die Datei bereits geöffnet ist.

Beispiel 2

```
STEP1:
FILE_OPEN (fc~b, "test.dat");
STEP2:
WHILE (NOT fc~b.EOF) DO
    FILE_READ (fcb, struct1);
END_WHILE;
```

Wird während der WHILE-Schleife ein Online-Editieren durchgeführt, dann wird STEP2 rückgesetzt und der Befehl FILE_READ funktioniert ordnungsgemäß. In diesem Fall wird wahrscheinlich ein Zeitproblem auftreten, da FILE_READ nicht abgebrochen wurde und immer noch aktiv sein kann, wenn das Online-Editieren ausgeführt wird.

Strukturierte Textprogramme

- Strukturierter Text unterstützt das nahtlose Online-Editieren nicht. Wurden bei laufendem Programm Änderungen an der strukturierten Textdatei durchgeführt, dann sind beim Speichern der Datei die Schaltflächen für nicht nahtloses Online-Editieren freigegeben.

Anweisungslistenprogramme

- Wurden bei laufendem Programm Änderungen an der Anweisungslistendatei durchgeführt, dann sind beim Speichern der Datei die Schaltflächen für nicht nahtloses Online-Editieren freigegeben.

Kapitel 5

Ablauf von Anwendungsprogramme

Nachdem Sie ein Anwenderprogramm erstellt haben, müssen Sie das Runtimesystem starten, damit Sie die Anwendung ausführen und beobachten können.

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Starten des Runtimesystems
- Ausführen des Anwenderprogramms
- Konfiguration von Programmen für die automatische Ausführung
- Überwachung des Stromflusses
- Status eines Anwenderprogramms anzeigen

5

Runtime-Subsysteme

Die Runtime-Subsysteme bestehen aus den Subsystemen Programm-Manager, Programmausführung, E/A-Zyklussteuerung und Ereignisprotokoll. Die Subsysteme Programm-Manager, Programmausführung und E/A-Zyklussteuerung werden optisch dargestellt durch das PC Control Symbol. Das Ereignisprotokoll hat sein eigenes Symbol.

Die Subsysteme Programmausführung und E/A-Zyklussteuerung haben Echtzeit-Prozeßpriorität. Das heißt, daß ihnen vor allen normalen Anwendungen (einschließlich Mausaktualisierung und Plattenzugriff) CPU-Zeit zugeteilt wird.

Runtime-Subsystem starten:

Starten Sie zunächst den Programmeditor und führen dann einen der folgenden Schritte durch:

- Wählen Sie *Runtime-Subsysteme starten* im Menü *Ausführen* des Programmeditors oder Bedienerschnittstellen-Editors.
- Wählen Sie *Runtimemaschine* aus dem Menü *PC Control Anwendungen* im Windows-Menü *Start*.

Ein einzelnes Programm ausführen

Um ein Programm ausführen zu können, muß das Windows NT Betriebssystem aktiv sein und es muß die PC Control Software installiert sein. Sind die Runtime-Subsysteme nicht aktiv, erscheint die Aufforderung, die Runtime-Subsysteme zu starten.

Das aktive Programm ausführen

Um das aktive Programm auszuführen:

1. Klicken Sie auf *ausführen.*
2. Wählen Sie *Ausführen*.

Nachdem das Programm mit der Ausführung beginnt, wird die Programmanzeige mit den aktiven Hervorhebungsfarben hervorgehoben, um den Status des laufenden Programms anzuzeigen.

Aktive Datei mit freigegebenem Austesten ausführen:

1. Klicken Sie auf *Ausführen.*
2. Wählen Sie den Menübefehl *RUN mit Austesten*.

Einen Schritt der aktiven Datei ausführen:

1. Klicken Sie auf *Ausführen.*
2. Wählen Sie *Einzelschritt*. Enthält der Schritt mehr als eine Befehlszeile, dann wird die nächste Befehlszeile in dem Schritt ausgeführt.

Die aktive Datei mit Neustart ausführen:

1. Klicken Sie auf *Ausführen.*
2. Wählen Sie *RUN mit Neustart*. Ein mit Neustart laufendes Programm beginnt automatisch seine Ausführung, wenn die Runtime-Subsysteme hochlaufen.

Ein laufendes Programm abbrechen

Um eine laufende Ablaufkette abzubrechen:

1. Klicken Sie auf *Ausführen.*
2. Wählen Sie *Abbrechen*. Das Programm wird abgebrochen und auf den Programmstart rückgesetzt.

Hinweis

Um in einem Ablaufketten-Schritt eine Programmunterbrechung herbeizuführen, verwenden Sie in diesem Schritt die BREAK-Funktion im strukturierten Text.

Programme für automatischen Ablauf konfigurieren

In einigen Anwendungen muß bei jedem Einschalten der Steuerung oder bei jedem hochlaufen der Runtime-Subsysteme eine Ablaufkette automatisch gestartet werden. Die erste Option können Sie mit einer Batchdatei bewerkstelligen, die die entsprechenden Befehle enthält und im Windows NT Startordner hinzugefügt wird. Die zweite Option wird erfüllt, indem der Befehl RUN mit Neustart verwendet wird.

Programme mit einer Batchdatei starten

Hinweis

Mit dieser Methode kann nur ein AS-Anwenderprogramm gestartet werden.

1. Öffnen Sie in Windows NT einen einfachen Texteditor, um eine Textdatei zu erstellen. Speichern Sie diese Datei mit der Dateinamenerweiterung .BAT. Tragen Sie die beiden folgenden Zeilen in diese Textdatei ein:

```
start c:\CIMPLICITY PC Control \bin\runtime.exe/RUN
          c:\CIMPLICITY PC Control \MyProject\Main.sst

start c:\CIMPLICITY PC Control \bin\oicfg.exe
```

Hinweis

Der vorstehende Befehl /RUN muß in GROSSBUCHSTABEN eingegeben werden. Mit dieser Methode kann nur ein AS-Anwenderprogramm gestartet werden.

2. Der Pfad zum Verzeichnis \bin muß den Gegebenheiten in Ihrer Maschine entsprechen.
3. Stellen Sie sicher, daß der Pfad zu Ihrem Projektordner und der Name der Ablaufkette stimmen. Verwenden Sie für die AS-Datei die Dateinnamenerweiterung .SST, nicht .SFC. Hier handelt es sich nicht um die binäre AS-Datei, die der Programmeditor benötigt, sondern um die strukturierte Textversion der AS-Datei, die der Runtimecompiler benötigt.
4. Stellen Sie bei der Verwendung des Programmeditors sicher, daß das aktuelle Projekt und die aktive Konfiguration für diese Anwendung korrekt eingestellt sind. Führen Sie bei der AS-Datei eine Syntaxanalyse durch um sicherzustellen, daß die entsprechende .SST-Datei existiert und aktuell ist.
5. Verwenden Sie das Bedienerschnittstellen-Dienstprogramm, um die entsprechende beim Einschalten zu verwendende .OPI-Datei auszuwählen.

6. Befolgen Sie die Anweisungen in Ihren Windows NT Hilfedateien, wenn Sie die Batchdatei in Ihrem NT Startordner hinzufügen.
7. Booten Sie Ihr System neu zum Testen.

Programme mit dem Befehl "RUN mit Neustart" starten

Die aktive Datei mit Neustart ausführen:

1. Klicken Sie auf *Ausführen.*
2. Wählen Sie den Menübefehl *RUN mit Nestart*. Ein mit Neustart ausgeführtes Programm läuft automatisch an, wenn die PC CONTROL Runtime-Subsysteme hochlaufen. Ein abgebrochenes oder fehlerhaftes Programm wird nicht mehr mit Lauf mit Neustart markiert.

Nachdem das Programm mit der Ausführung beginnt, wird die Programmanzeige mit den aktiven Hervorhebungsfarben hervorgehoben, um den Status des laufenden Programms anzuzeigen.

Stromfluß überwachen

Wenn das Programm läuft, wird die Programmanzeige mit den aktiven Hervorhebungsfarben hervorgehoben, um den Status des laufenden Programms anzuzeigen.

Aktive Kontaktplanprogramme

Läuft ein Kontaktplanprogramm oder sind in einer laufenden Ablaufkette eine eingebettete Aktion und/oder ein Kontaktplantransitionsprogrammfenster geöffnet, dann werden die Kontakte, Spulen und Funktionsblöcke in dem Programm optisch hervorgehoben. Ein Schließerkontakt wird hervorgehoben, wenn das mit ihm verknüpfte Boolesche Symbol TRUE ist. Ein Öffnerkontakt wird hervorgehoben, wenn das mit ihm verknüpfte Boolesche Symbol FALSE ist. Ein Kontakt zur Erkennung positiver Übergänge wird hervorgehoben, wenn an dem verbundenen Symbol ein positiver Übergang stattfindet. Ein Kontakt zur Erkennung negativer Übergänge wird hervorgehoben, wenn an dem verbundenen Symbol ein negativer Übergang stattfindet. Alle Ausgangsspulen werden hervorgehoben, wenn das verbundene Boolesche Symbol TRUE ist (die Hervorhebung der Spule gibt den Zustand des Spulensymbols an). Alle Funktionsblöcke werden hervorgehoben, wenn der Funktionsblock aktiv ist.

Hinweis

Ist das mit einem Kontakt verknüpfte Boolesche Symbol TRUE, dann wird der Kontakt unabhängig von seiner Position im Strompfad hervorgehoben. Das heißt, daß die Kontakte vor einem hervorgehobenen Kontakt nicht unbedingt auch TRUE sein müssen.

Aktive Ablaufketten

Läuft eine Ablaufkette, dann werden alle aktiven Schritte und/oder Transitionen hervorgehoben. Wird ein aktiver Schritt mit Prozeßbefehlen angezeigt, dann erscheint links von dem gerade ausgeführten Befehl eine aktive Befehlsanzeige.

Status von Anwenderprogrammen anzeigen

Um den Programmstatus anzuzeigen:

1. Klicken Sie auf *Anzeigen.*
2. Wählen Sie *Programmstatus*. Hierauf erscheint das Programmstatusfenster mit einer Liste der aktiven Programme und deren Status.

Um das Programm anzuzeigen:

Doppelklicken Sie auf ein Programm.

Um einen Programmbefehl auszugeben:

Wählen Sie das gewünschte Programm aus dem Listenfeld und drücken Sie die Taste mit dem gewünschten Programmbefehl: Anzeigen, Ausführen, Abbrechen, Stop oder Schritt.

Kapitel 6

Erstellen von Bedienerschnittstellen-Anwendungen

PC Control enthält eine integrierte Bedienerschnittstellensoftware (PC Control GBO), die zum Bedienen und Beobachten von Automatisierungsanwendungen konfiguriert werden kann. Durch die enge Integration mit dem Programmeditor sind alle globalen Symbole in der aktiven Konfiguration in der Bedienerschnittstellensoftware unmittelbar verfügbar.

In diesem Kapitel finden Sie Informationen zu folgenden Themen:

- Übersicht über Bedienerschnittstellensoftware
- Erstellen eines Bedienerschnittstellenmenüs
- Symboloperationen
  - Symbole bearbeiten
  - Konfigurationen aktivieren
- Erstellen von Bedienerschnittstellen-Anwendungen
  - Standard-Bedienelemente
  - ActiveX-Bedienelemente

Informationen zu Bewegungssteuerungen finden Sie in Anhang E.

6

Abschnitt 1: Bedienerschnittstelleneditor - Übersicht

Bedienerschnittstelle starten

Mit der Bedienerschnittstellensoftware können Sie Bedienerschnittstellen für eine Anwendung erstellen und die Anlagenbediener die von Ihnen erzeugte Schnittstelle benutzen lassen.

Bedienerschnittstellensoftware starten:

- Suchen Sie das Menü *PC Control Anwendungen* im Windows-Menü *Start*, und wählen die Option *Bedienerschnittstelle*.
- Ist der Programmeditor offen, wählen Sie *Bedienerschnittstelle* aus dem Programmeditormenü *Werkzeuge*.

Die Bedienerschnittstelle startet im Aktivierungsmodus. Die letzte für das aktuelle Projekt geöffnete Bedienerschnittstellendatei wird zur Definition der Bedienerschnittstelle verwendet. Das in der Bedienerschnittstellendatei definierte Startmenü wird als erstes Bedienerschnittstellenmenü angezeigt.

Laufen die Runtime-Subsysteme nicht, wenn die Bedienerschnittstelle gestartet wird, erscheint eine Aufforderung, die es Ihnen ermöglicht, die Runtime-Subsysteme zu starten.

Zugriffsebenen

Über Zugriffsebenen und Zugangscodes wird der unbefugt Zugriff auf Anwenderprogramme und Konfigurationsdaten verhindert. Es gibt fünf verschiedene Zugriffsebenen (0-4), jede mit einem eigenen Zugangscode.

In der nachstehenden Tabelle werden die Privilegien der einzelnen Ebenen beschrieben.

Zugriffsebene:	Sie können ...
0	Bedienermenüs und Bedienelemente aktivieren. Ablaufketten auswählen und laufen lassen.
1	Kontinuierliche Kontaktplanprogramme für Umgebungs- und Maschinenlogik laufenlassen und abbrechen. Kontaktplanprogramme und Ablaufketten anzeigen.
2	Ablaufketten bearbeiten. Projektmanagementfunktionen ausführen.
3	Kontaktplanprogramme editieren. Bedienerschnittstelle modifizieren. Systemkonfigurationsdateien modifizieren.
4	Passworte für Zugriffsebenen 1 bis 4 verändern.

Die Standardwerte der Zugangscodes finden Sie auf den gedruckten Installationsanweisungen, die der PC CONTROL Software bei Lieferung beiliegen.

Zugangscode eingeben

Programmeditor und Bedienerschnittstelle benötigen Zugangscodes. Der Programmeditor zeigt beim Hochlaufen automatisch das Zugriffsebenen-Tastenfeld an.

Zugangscode eingeben:

In:	... dann ...
Programmeditor	auf die Zahlen des Tastenfeldes klicken, die Ihrem Zugangscode entsprechen, und dann auf *OK* klicken.
Bedienerschnittste lle	in der Menüleiste auf *Zugang* klicken und *Passwort* auswählen. Hierauf erscheint ein Tastenfeld. auf die Zahlen des Tastenfeldes klicken, die Ihrem Zugangscode entsprechen, und dann auf *OK* klicken.

Zugriffsebene auf 0 einstellen:

Klicken Sie im Zugangs-Tastenfeld auf die Schaltfläche *Aufheben*.

Zugriffsebene wechseln

Der Zugangscode für die einzelnen Ebenen kann nur über die Bedienerschnittstelle mit einem Zugangscode für Ebene 4 verändert werden. Für jede Ebene gibt es einen eigenen Zugangscode.

Zugriffsebenen-Passwort verändern:

1. In der Menüleiste auf *Zugang* klicken und *Passwort* auswählen. Hierauf erscheint ein Tastenfeld.
2. Klicken Sie viermal auf die Taste '*'. Hierauf erscheint die Meldung "Geben Sie die Zugriffsebene ein, die Sie ändern wollen".
3. Klicken Sie auf die Nummer der Zugriffsebene, die Sie ändern wollen, und dann auf *OK*.
4. Klicken Sie auf die vier neuen Ziffern. Hierauf erscheint die Meldung "Neues Passwort nochmals eingeben".
5. Klicken Sie nochmals auf die vier Ziffern des Passwortes und anschließend auf *OK*. Haben Sie das Passwort richtig bestätigt, wird es geändert.

Hinweis

Klicken Sie im Tastenfeld auf AUFHEBEN, um die Zugriffsebene auf Null einzustellen.

Werden Steuerfunktionen mit Bedienelementen verknüpft (im Bedienerschnittstellen-Editiermodus), können Sie eine Zugriffsebene angeben, mit der die Benutzung der einzelnen Bedienelemente geregelt wird.

Aktivierungs- und Editierbetriebsarten

Die Bedienerschnittstellensoftware enthält eine Aktivierungs- und Editierbetriebsart. Im Aktivierungsmodus läuft die Bedienerschnittstelle und steuert die Anwendungsfunktionen. Im Editiermodus können Sie Menüs bearbeiten, erstellen, kopieren, umbenennen oder löschen, diese Menüs mit Bedienelementen versehen und Bedienelemente mit Programmen und Symbolen verbinden. Zum Aufruf des Editiermodus benötigen Sie Zugriffsebene 3 oder 4.

Die Bedienerschnittstelle wird im Aktivierungsmodus gestartet.

Bedienerschnittstellen-Betriebsarten umschalten

Die Bedienerschnittstelle wird im Aktivierungsmodus gestartet.

Aus Aktivierungsmodus in Editiermodus umschalten:

Zum Erstellen und Bearbeiten von Bedienerschnittstellenmenüs benötigen Sie Zugriff auf Ebene 3 oder 4.

1. Wählen Sie im Werkzeugmenü die Option "Bedienerschnittstellenmenü bearbeiten".
2. Geben Sie das Passwort für den Zugang zu Ebene 3 oder 4 ein.

Im Editiermodus können Sie Bedienelemente einfügen, bearbeiten, verschieben und in der Größe verändern.

Aus dem Editiermodus in den Aktivierungsmodus umschalten:

Klicken Sie auf "Ausführen" und wählen Sie dann die Option "Menüs aktivieren".

Nach der Aktivierung der Bedienermenüs können mit den Bedienelementen Anwenderfunktionen aktiviert und gesteuert werden. Sie können die Bedienelemente nicht bearbeiten, wenn die Menüs aktiviert sind.

6

Kontaktplanprogramme über Bedienerschnittstelle steuern

Der Logikmanager ist ein ablauffähiges Programm, mit dem Kontaktplanprogramme in einem Projekt gesteuert werden können. Um den Logikmanager aus einem Bedienerschnittstellenmenü heraus aufrufen zu können, fügen Sie eine Schaltfläche hinzu und hängen die Funktion "AUSFÜHREN" daran. Wählen Sie den Logikmanager als ausführbare Datei, die ablaufen soll. Weitere Angaben finden Sie unter "Schaltflächen”.

Zur Laufzeit zeigt der Logikmanager eine Liste der Kontaktplanprogramme, die im aktiven Projekt gerade laufen, sowie eine Liste von Kontaktplanprogrammen, die im aktiven Projekt gerade nicht laufen. Laufende Kontaktplanprogramme werden in der Liste *Laufende Kontaktplanprogramme* angezeigt. Nicht laufende Kontaktplanprogramme werden in der Liste *Projekt-Kontaktplanprogramme* angezeigt.

Ein Kontaktplanprogramm ausführen

1. Wählen Sie das Programm aus der Liste *Projekt-Kontaktplanprogramme* aus.
2. Klicken Sie auf *Ausführen*. Das ausgewählte Programm wird aus der Liste *Projekt-Kontaktplanprogramme* entfernt und in die Liste *Laufende Kontaktplanprogramme* eingetragen.

Ausführen eines Kontaktplanprogramms mit Neustart

1. Wählen Sie das Programm aus dem Listenfeld *Projekt-Kontaktplanprogramme*.
2. Drücken Sie die Schaltfläche *RUN mit Neustart*. Das ausgewählte Programm wird aus der Liste *Projekt-Kontaktplanprogramme* entfernt und in die Liste *Laufende Kontaktplanprogramme* eingetragen. Ein mit Neustart ausgeführtes Programm läuft automatisch an, wenn die Runtime-Subsystemsoftware hochläuft.

Abbrechen eines Kontaktplanprogramms

1. Wählen Sie das Programm aus der Liste *Laufende Kontaktplanprogramme* aus.
2. Drücken Sie *Abbrechen*. Das ausgewählte Programm wird aus dem Listenfeld *Laufende Kontaktplanprogramme* entfernt und in die Liste *Projekt-Kontaktplanprogramme* eingetragen.

Ein Kontaktplanprogramm bearbeiten

1. Wählen Sie das Programm aus der Liste *Projekt-Kontaktplanprogramme* oder aus der Liste *Laufende Kontaktplanprogramme* aus.
2. Klicken Sie auf *Editieren*. Der Programmeditor wird aktiviert, das ausgewählte Kontaktplanprogramm steht vorne. Läuft das Kontaktplanprogramm gerade, dann hebt der Programmeditor das laufende Kontaktplanprogramm hervor.

Abschnitt 2: Arbeiten mit Bedienerschnittstellenmenüs

Bedienerschnittstellen-Operationen

Starten einer neuen Bedienerschnittstellendatei

Eine Bedienerschnittstellendatei enthält ein oder mehrere Menüs. Ein Projekt kann mehrere Bedienerschnittstellendateien enthalten. Das Anfangsmenü der zuletzt geöffneten Bedienerschnittstellendatei ist das aktive Menü beim Start der Bedienerschnittstelle.

Starten einer neuen Bedienerschnittstellendatei

- Wählen Sie *Neu* aus dem Menü *Datei* oder benutzen Sie die Schaltfläche in der Symbolleiste.

Der Standard-Dateiname einer Bedienerschnittstellendatei lautet *OPI*, der Standard-Menüname ist *Bedienerschnittstelle*.

Öffnen einer Bedienerschnittstellendatei

Öffnen einer neuen Bedienerschnittstellendatei

- Wählen Sie *Öffnen* aus dem Menü *Datei* oder benutzen Sie die Schaltfläche in der Symbolleiste.

Speichern einer Bedienerschnittstellendatei

Beim Speichern einer Bedienerschnittstellendatei werden alle Menüs in der Datei gespeichert.

Speichern der Datei

- Wählen Sie *Speichern* aus dem Menü *Datei* oder benutzen Sie die Schaltfläche in der Symbolleiste.

Beim erstmaligen Speichern einer Datei erscheint die Dialogbox *Speichern als*, in der Sie der Datei einen Namen zuweisen können.

Datei unter neuem Namen oder in einem neuen Ordner speichern

- Wählen Sie *Speichern als* im Menü *Datei*. Geben Sie in der aufgeblendeten Dialogbox für die Datei einen neuen Ordner und/oder Namen an.

6

Menüoperationen

Erstellen eines neuen Bedienerschnittstellenmenüs

Erstellen eines neuen Bedienerschnittstellenmenüs

1. Wählen Sie *Neues Menü* aus dem Menü *Bearbeiten*. Hierauf erscheint die Dialogbox *Neuen Menünamen eingeben*.
2. Geben Sie den Namen des neuen Menüs ein und klicken dann auf *OK*. Ein leeres Bedienermenü wird angelegt, beidem der Menüname in der Kopfleiste steht.

Menü löschen

Die Bedienerschnittstellendatei muß mindestens ein Menü enthalten. Eine Fehlermeldung wird ausgegeben, wenn Sie versuchen, das letzte Menü der Bedienerschnittstelle zu löschen.

Ein Bedienerschnittstellenmenü löschen

1. Wählen Sie *Menü löschen* aus dem Menü *Bearbeiten*. Hierauf erscheint die Dialogbox *Menü zum Löschen auswählen*.
2. Geben Sie den Namen des Menüs ein, das Sie löschen wollen, oder selektieren Sie den Menünamen in dem Drop-Down-Listenfeld und klicken dann auf *OK*.

Wenn Sie das Startmenü gelöscht haben, dann erscheint die Dialogbox *Neues Startmenü wählen*. Geben Sie den Namen des neuen Startmenüs ein oder wählen Sie den den Menünamen in dem Drop-Down-Listenfeld aus.

Menü kopieren

Ein Bedienerschnittstellenmenü kopieren

1. Wählen Sie *Menü kopieren* aus dem Menü *Bearbeiten*. Hierauf erscheint die Dialogbox *Menü zum Kopieren auswählen*.
2. Wählen Sie das Menü aus der Dialogbox und klicken Sie dann auf *OK*. Hierauf erscheint die Dialogbox *Neuen Menünamen eingeben*.
3. Geben Sie den Namen des neuen Menüs ein und klicken dann auf *OK*. Das neue Menü wird nun mit dem gleichen Inhalt wie das ursprüngliche Menü erstellt.

Menü umbenennen

Ein Bedienerschnittstellenmenü umbenennen

1. Wählen Sie *Menü umbenennen* aus dem Menü *Bearbeiten*. Hierauf erscheint die Dialogbox *Menü zum Umbenennen auswählen*.
2. Wählen Sie das Menü, das Sie umbenennen wollen, und klicken dann auf *OK*. Hierauf erscheint die Dialogbox *Neuen Menünamen eingeben*.
3. Geben Sie den Namen des neuen Menüs ein und klicken dann auf *OK*.

Startmenü auswählen

Das Startmenü ist das Menü, das beim Starten der Bedienerschnittstelle erscheint (für die aktive Konfiguration des Projektes).

Startmenü auswählen

1. Wählen Sie *Startmenü* aus dem Menü *Bearbeiten*. Hierauf erscheint die Dialogbox *Neues Startmenü auswählen*.
2. Wählen Sie ein Startmenü und klicken dann auf *OK*. Das ausgewählte Menü ist das neue Startmenü.

Ein Menü zur Bearbeitung auswählen

Sie müssen ein Menü zuerst auswählen, ehe Sie es bearbeiten können.

Ein Bedienermenü zur Bearbeitung auswählen

1. Klicken Sie auf das Menüelement *Menüs*. Im Menü erscheint eine Liste aller in der aktuellen Bedienerschnittstellendatei enthaltenen Bedienermenüs.
2. Wählen Sie das gewünschte Menü aus. Das Menü wird zur Bearbeitung angezeigt.

Abschnitt 3: Mit Bedienelementen arbeiten

Bedienelemente hinzufügen

Ein Bedienelement hinzufügen

1. Wählen Sie *Neues Standard-Bedienelement* aus dem Menü *Bearbeiten*. Hierauf erscheint eine Liste der Standard-Bedienelemente.
2. Wählen Sie das Bedienelement, das Sie hinzufügen wollen. Die Cursorform wechselt zur Anzeige des Bedienelements.
3. Setzen Sie den Cursor in das Menü und klicken Sie, um das Bedienelement einzufügen.

Hinweis

Die meisten Bedienelemente können Sie auch aus der Bedienelemente-Symbolleiste auswählen.

Drücken Sie Esc, um den Vorgang rückgängig zu machen.

Information über das Einfügen von ActiveX-Bedienelementen finden Sie unter "Einfügen von ActiveX-Bedienelementen."

Bedienelemente bearbeiten

Ein Standard-Bedienelement bearbeiten

1. Führen Sie einen der folgenden Schritte durch:
   - Doppelklicken Sie auf das Bedienelement.
   - Wählen Sie das gewünschte Bedienelement und drücken Sie dann die Eingabetaste.
   - Selektieren Sie das gewünschte Bedienelement wählen Sie dann *Standard-Bedienelement bearbeiten* aus dem Kontextmenü oder dem Menü *Bearbeiten*.
2. Hierauf erscheint eine für das Bedienelement geeignete Bearbeitungs-Dialogbox. Informationen zu Standard-Bedienelementen finden Sie unter "Standard-Bedienelemente".

Hinweis

Informationen zur Bearbeitung von ActiveX-Bedienelementen finden Sie unter "ActiveX-Bedienelemente bearbeiten" im Abschnitt "ActiveX-Bedienelemente".

Bedienelemente auswählen

Auswahl eines einzelnen Bedienelements

- Setzen Sie den Mauscursor auf das Bedienelement und klicken die Maustaste. Das Bedienelement wird hervorgehoben.

Mehrere Bedienelemente über eine Auswahlbox auswählen

Um die auszuwählenden Bedienelemente wird ein rechteckiges *Gummiband* gezogen.

1. Setzen Sie den Auswahlpfeil auf eine Ecke des Rechtecks. Klicken Sie und ziehen den Pfeil zur gegenüberliegenden Ecke.

2. Lassen Sie die Maustaste los. Alle Bedienelemente, die vollständig in dem rechteckigen Bereich liegen, werden selektiert.

Mehrere Bedienelemente mit Tastatur und Maus auswählen

- Halten Sie beim Selektieren der einzelnen Bedienelemente die Taste Ctrl oder Shift gedrückt.

Ein Bedienelement aus einer Gruppe ausgewählter Bedienelemente abwählen

- Halten Sie die Taste Ctrl oder Shift gedrückt, während Sie den Mauscursor auf das Bedienelement setzen und klicken.

Bedienelemente verschieben

Bedienelemente mit der Maus verschieben

1. Wählen Sie die Bedienelemente.

2. Setzen Sie den Cursor auf eines der selektierten Bedienelemente. Klicken Sie und ziehen das selektierte Bedienelement an die gewünschte Stelle. Ein schwarzer Rahmen zeigt den neuen Bereich der Bedienelementegruppe.

3. Lasen Sie die Maustaste los, um die Bedienelemente an der neuen Stelle zu plazieren.

   Drücken Sie die Taste Esc, um den Vorgang rückgängig zu machen..

Bedienelemente mit der Tastatur verschieben

1. Wählen Sie die Bedienelemente.

2. Verschieben Sie die Bedienelementegruppe mit den Pfeiltasten um einen Bildpunkt in Pfeilrichtung.

Größe der Bedienelemente verändern

Größe eines Bedienelements mit der Maus verändern

1. Wählen Sie die Bedienelemente. Kann die Größe des Bedienelements verändert werden, erscheinen an den Ecken und Seiten des Bedienelements Ziehpunkte.
2. Setzen Sie den Cursor auf einen der Ziehpunkte. Der Cursor wechselt in einen Größeneinstellungspfeil, der die Richtung anzeigt, in der die Größe des Bedienelements verändert werden kann.
3. Klicken Sie auf den gewünschten Ziehpunkt und ziehen Sie ihn so, daß das Bedienelement wie gewünscht vergrößert oder verkleinert wird. Ein schwarzer Rahmen zeigt den neuen Bereich der Bedienelementegruppe.
4. Lassen Sie die Maustaste los, wenn Sie die Größe des Bedienelements eingestellt haben.

Drücken Sie die Taste Esc, um den Vorgang rückgängig zu machen..

Kopieren, Ausschneiden und Einfügen

Die Arbeitsgänge Kopieren, Ausschneiden und Einfügen werden oft zusammen verwendet. Kopieren speichert die Auswahlen in der Zwischenablage. Ausschneiden löscht die Auswahl und speichert sie in der Zwischenablage. Einfügen fügt den Inhalt der Zwischenablage in das aktuelle Menü ein. Sie können auch zwischen verschiedenen Menüs ausschneiden, kopieren und einfügen.

Obwohl die Arbeitsgänge Kopieren, Ausschneiden und Einfügen ähnlich wie bei anderen Windows-Anwendungen funktionieren, ist das Zwischenablageformat der Bedienelemente zu anderen Anwendungen nicht kompatibel. Bedienelemente können nur innerhalb oder zwischen Menüs kopiert und eingefügt werden.

Ein Bedienelement kopieren

- Selektieren Sie das Bedienelement und wählen dann *Kopieren* aus dem Menü *Bearbeiten* oder dem Kontextmenü, oder benutzen Sie die Tasten Ctrl+C auf der Tastatur.

Ein Bedienelement ausschneiden

- Selektieren Sie das Bedienelement und wählen dann *Ausschneiden* aus dem Menü *Bearbeiten* oder dem Kontextmenü, oder benutzen Sie die Tasten Ctrl+X auf der Tastatur.

Inhalt der Zwischenablage einfügen

1. Rufen Sie das Menü auf, in das Sie das Bedienelement einfügen wollen (falls dies nicht das aktuelle Menü ist).
2. Wählen Sie *Einfügen* aus dem Menü *Bearbeiten* oder dem Kontextmenü, oder benutzen Sie die Tasten Ctrl+V auf der Tastatur.

Die Bedienelement werden aus der Zwischenablage heraus im Bedienermenü abgelegt. Sie erscheinen in der Position, die Sie beim Ausschneiden oder Kopieren belegten.

Bedienelemente löschen

Bedienelemente aus dem Bedienermenü löschen

1. Wählen Sie die Bedienelemente.
2. Drücken Sie die Taste Del oder wählen Sie *Löschen* aus dem Menü *Bearbeiten* oder dem Kontextmenü.

Bedienelemente ausrichten

Sie können Bedienelemente nach links, rechts, oben oder unten ausrichten.

Links	Die ausgewählten Bedienelemente werden so ausgerichtet, daß ihre linke Seite mit dem am weitesten links stehenden Bedienelement bündig ist.
Rechts	Die ausgewählten Bedienelemente werden so ausgerichtet, daß ihre rechte Seite mit dem am weitesten rechts stehenden Bedienelement bündig ist.
Oben	Die ausgewählten Bedienelemente werden so ausgerichtet, daß ihre Oberkante mit dem am weitesten oben stehenden Bedienelement bündig ist.
Unten	Die ausgewählten Bedienelemente werden so ausgerichtet, daß ihre Unterkante mit dem am weitesten unten stehenden Bedienelement bündig ist.

Bedienelemente ausrichten

1. Selektieren Sie die Bedienelemente, die Sie ausrichten wollen (Sie müssen mindestens zwei Bedienelemente selektieren).
2. Wählen Sie *Bedienelement ausrichten* aus dem Menü *Layout*, und wählen Sie dann die gewünschte Ausrichtung: *links*, *rechts*, *oben* oder *unten*.

Bedienelement nach hinten/vorne schieben

Bei einander überlappenden Bedienelementen können Sie ein Bedienelement nach hinten oder nach vorne schieben.

Ein Bedienelement nach vorne oder hinten schieben

1. Wählen Sie das Bedienelement aus.
2. Führen Sie einen der folgenden Schritte durch:
   - Um das selektierte Bedienelement vor alle anderen überlappenden Bedienelemente zu schieben, wählen Sie im Menü *Layout* die Option *Nach vorne*.
   - Um das selektierte Bedienelement hinter alle anderen überlappenden Bedienelemente zu schieben, wählen Sie im Menü *Layout* die Option *nach hinten*.

Abschnitt 4: Symboloperationen

Der Symbolmanager kann direkt im Bedienerschnittstelleneditor verwendet werden. Der Symbolmanager darf jedoch nicht gleichzeitig im Bedienerschnittstelleneditor und im Programmeditor geöffnet sein. Wird der Symbolmanager vom Bedienerschnittstelleneditor aus geöffnet, dann wirkt er nur auf globale Symbole. Er ist verfügbar von der Symbolleiste, dem Bedienerschnittstellenmenü und von Bedienelemente-Dialogboxen, bei denen Sie Symbole auswählen können.

Hinweis

In den Bedienerschnittstellen-Bedienelementen können nur globale Symbole verwendet werden.

Symbole bearbeiten

Globale Symbole bearbeiten

- Wählen Sie *Symbolmanager* aus dem Menü *Werkzeuge*. Weitere Angaben zur Benutzung des Symbolmanagers finden Sie unter **Symbolmanager** im **Programmierleitfaden**.

Konfigurationen aktivieren

Sie müssen die Konfiguration aktivieren, damit die bearbeiteten Symbole im Programmeditor verfügbar werden.

Führen Sie zur Aktivierung der Konfiguration einen der folgenden Schritte durch:

- Klicken Sie auf *Anwenden* im Symbolmanager .
- Wählen Sie *Konfiguration aktivieren* im Menü *Werkzeuge*.

Konfiguration speichern.

- Wählen Sie *Konfiguration speichern* im Menü *Werkzeuge*.

Abschnitt 5: Erstellen von Bedienerschnittstellen-Anwendungen

Standard-Bedienelemente

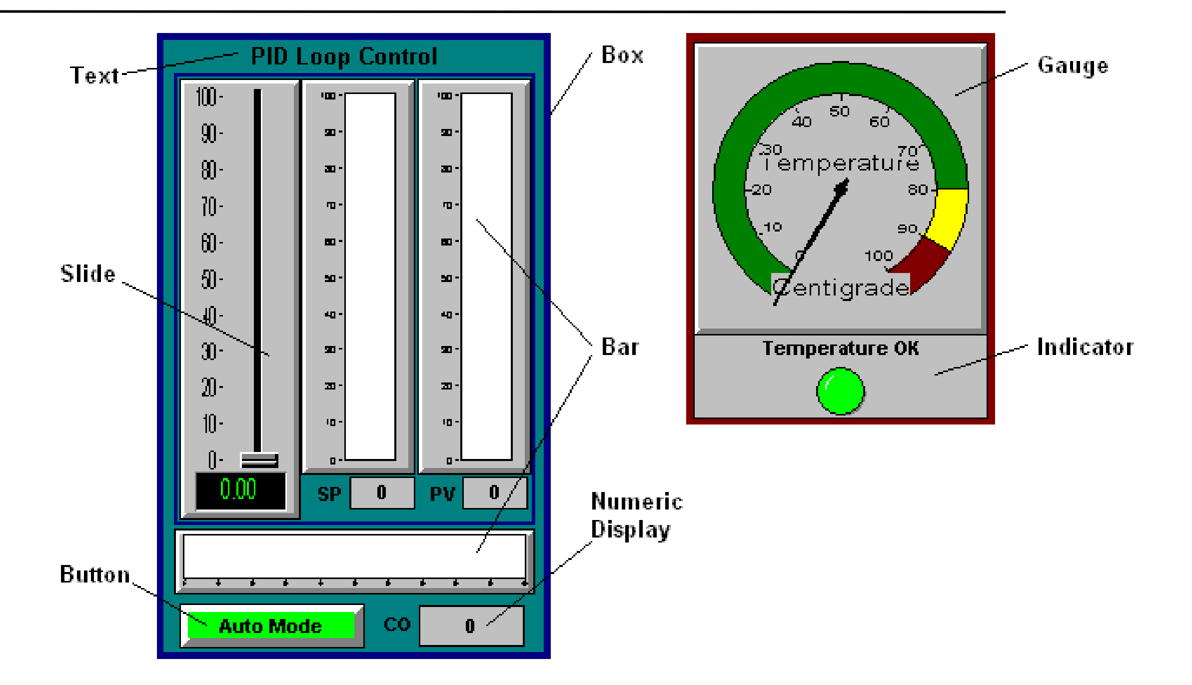
Einleitung

Nachstehend finden Sie eine Auflistung und kurze Beschreibung der Standard-Bedienelemente und Anzeigeelemente, die Sie in einem Bedienerschnittstellenmenü verwenden können.

Bedienelement	Beschreibung
Balken	Beim Balken werden eine Skala und ein Gleitbalken angezeigt. Der Gleitbalken zeigt den Verlauf des aktuellen Wertes des mit ihm verknüpften Symbols an. Der Balken kann horizontal oder vertikal ausgerichtet sein.
Bitmap	Eine Bitmap ist ein Anzeigeelement, mit dem das Anwendungsmenü beschrieben oder einem Menü optische Anreize hinzugefügt werden können.
Kästchen	Ein Kästchen ist ein Anzeigeelement, mit dem Bedienelemente zusammengefaßt, Text oder Bitmaps eingerahmt, Menüteile hervorgehoben oder Menüs auf andere Weise interessant gestaltet werden können.
Klicktaste (Mehrzustandstaste)	Diese Taste führt eine Funktion aus, wenn Sie auf sie klicken. Zu den Funktionen gehören ein Boolesches Symbol setzen oder rücksetzen, ein Menü anzeigen, ein Programm auswählen, laufen lassen, anhalten oder abbrechen, Bedienereingaben entgegennehmen, und eine ablauffähige Datei ausführen. Die Taste kann für Operationen mit einem Zustand oder mit zwei Zuständen mit Zustandsüberspeicherung konfiguriert werden. Für jeden Zustand können Funktionen programmiert werden und die Taste führt die Funktionen entsprechend ihrem aktuellen Zustand aus. Konfiguration ist möglich für automatische Zustandsänderung oder Zustandsänderung auf der Grundlage eines Booleschen Symbols.

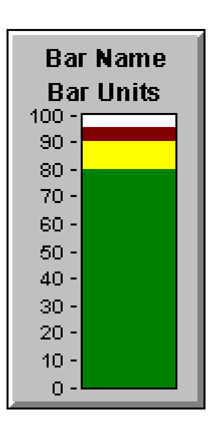
Drucktaste	Eine Drucktaste führt eine Funktion aus, wenn Sie auf sie klicken. Zu den Funktionen gehören ein Boolesches Symbol setzen oder rücksetzen, ein Menü anzeigen, ein Programm auswählen, laufen lassen, anhalten oder abbrechen, Bedienereingaben entgegennehmen, und eine ablauffähige Datei ausführen. Je nach Programmierung werden die Funktionen beim Drücken oder beim Loslassen der Drucktaste ausgeführt.
Anzeigeuhr	Die Anzeigeuhr besitzt eine Skala und einen drehbaren Zeiger. Der Drehzeiger gibt den aktuellen Wert eines der Uhr zugeordneten Symbols an. Es gibt vier verschiedene Uhrtypen.
Kennmelder	Ein Kennmelder ist eine Zustandssteuerung, deren Erscheinungsbild (Text, Farbe, usw.) sich entsprechend dem Wert eines Symbols vom Typ BOOL oder BYTE verändert.
Numerisch	Mit der numerischen Anzeige wird der numerische Wert eines Symbols angezeigt.
Auswahlprogramm-Statustafel	Die Auswahlprogramm-Statustafel zeigt Namen und aktuellen Status (z.B. läuft, gestoppt, usw.) des gewählten Programms an. Das Auswahlprogramm ist das Programm, das Sie über eine Klicktaste oder Drucktaste mit dem Befehl PROGRAMM AUSWÄHLEN ausgewählt haben.
Schieber	Mit dem Schieber können Sie einen Symbolwert kontinuierlich verändern. Wird der Schieber nach oben oder unten bewegt, dann verändert sich der Wert des dem Schieber zugeordneten Symbols.
Text	Mit dem Textelement können Sie Bedienelemente und Bedienelementgruppen beschriften, Menüfunktionen dokumentieren, oder sonstige Beschreibungen von Menüoperationen oder Menüfunktionen erzeugen.

Die nachstehende Abbildung zeigt ein Beispielmenü mit Standard-Bedienelementen und Anzeigeelementen.

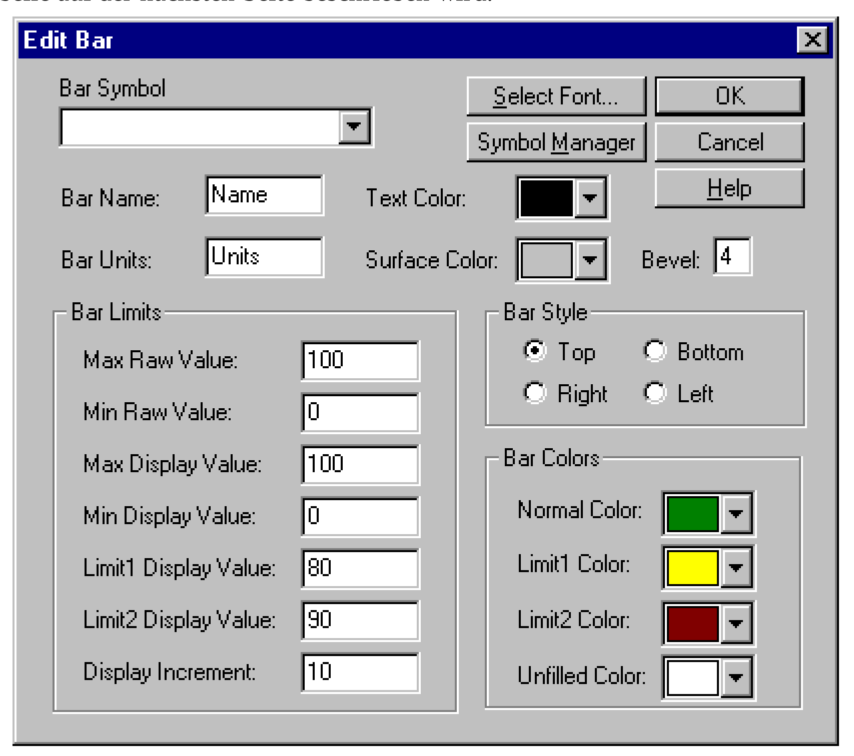

Text
PID Loop Control
Box
Slide
Bar
Button
Auto Mode
SP
PV
CO
Numeric
Display
Gauge
Temperature
Centigrade
Temperature OK
Indicator

Balken

Beim Balken werden eine Skala und ein Gleitbalken angezeigt. Der Gleitbalken zeigt den Verlauf des aktuellen Wertes des mit ihm verknüpften Symbols an. Bei einer Änderung des Symbolwertes wechselt die Farbe des Balkens und füllt den Balkenbereich entsprechend dem Symbolwert mehr oder weniger auf. Es können Grenzwerte eingestellt werden, so daß sich ein Farbumschlag ergibt, wenn sich der Symbolwert dem Grenzwert nähert. Der Balken kann horizontal oder vertikal ausgerichtet sein.



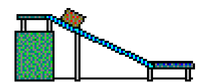
Die nachstehende Abbildung zeigt die Dialogbox *Balken bearbeiten*, die in der Tabelle auf der nächsten Seite beschrieben wird.



Feld	Beschreibung
Balkensymbol	Spezifiziert das Symbol, dessen Wert durch den Gleitbalken angezeigt wird. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*.
Balkennamen	Legt für das Bedienelement einen Namen fest. Nur zur Menüdokumentation.
Balkeneinheiten	Legt für das Bedienelement eine Einheit fest. Nur zur Menüdokumentation.
Textfarbe	Legt eine Farbe für Name, Einheiten und Skalennumerierung des Bedienelements fest.
Oberflächenfarbe	Gibt eine Hintergrundfarbe für das Bedienelement ein.
Schräge	Die Schräge hat Einfluß auf den Rahmen des Bedienelements. Die Werte der Schräge liegen zwischen 0 und 6. Eine Schräge von 0 läßt das Bedienelement flach erscheinen. Mit zunehmendem Wert wird die Darstellung des Bedienelements räumlicher.
Balkengrenzwerte	
Max. Rohwert Min. Rohwert	Dies sind die oberen und unteren Grenzwerte, auf die der Balken anspricht. Sie begrenzen den Wert des vom Balken dargestellten Symbols. Alle Werte, die größer als der maximale Rohwert sind, füllen den Gleitbalken bis zur Obergrenze. Alle Symbolwerte, die unterhalb des minimalen Rohwertes liegen, löschen den Gleitbalken. Wenn Sie die Werte von Max. Rohwert und Min. Rohwert vertauschen können Sie den invertierten Wert des Balkensymbols darstellen. Maximal- und Minimallogik sind vertauscht.
Max. Anzeigewert Min. Anzeigewert	Gibt die maximale und minimale Skalennumerierung an.
Grenze1 Anzeigewert Grenze2 Anzeigewert	Gibt Grenzwerte an, bei denen der Gleitbalken seine Farbe verändert (bezüglich Anzeigewerten). Zum Beispiel zur Anzeige von Alarmzuständen.
Anzeigeinkremente	Gibt das Skalennumerierungsinkrement an.
Balkenform	Die Balkenform gibt die Orientierung des Balkenelements sowie die Richtung an, in der sich der Balken füllt.
Balkenfarben	Es gibt vier verschiedene Balkenfarben: Normal - die Farbe des Balkens, wenn er den Wert anzeigt, aber noch keinen Grenzwert erreicht hat. Grenze 1 - die Farbe des Balkens, wenn er die angegebene Grenze 1 erreicht hat. Grenze 2 - die Farbe des Balkens, wenn er die angegebene Grenze 2 erreicht hat. Leer - die Hintergrundfarbe des Gleitbalkens.

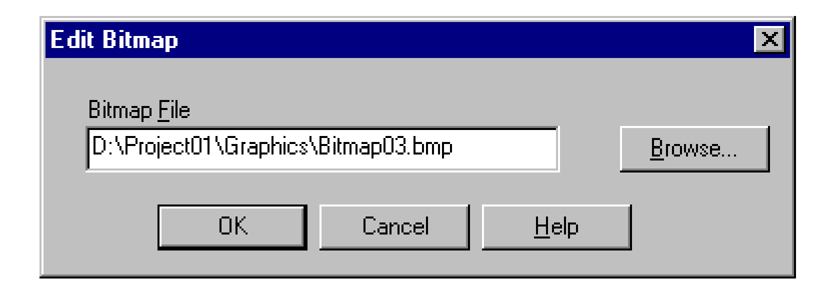
Feld	Beschreibung
Schriftart auswählen	Wenn Sie auf diese Schaltfläche klicken, erscheint eine Dialogbox *Schriftart*, mit der Sie Schriftart, Stil, Größe und Farbe des Bedienelementetexts (Name, Einheiten und Skalennumerierung) auswählen können.
Symbolmanager	Klicken Sie auf diese Schaltfläche, um den Symbolmanager zu öffnen.

Bitmap



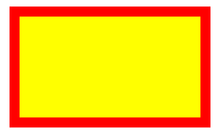
Eine Bitmap ist ein Anzeigeelement, mit dem das Anwendungsmenü beschrieben oder einem Menü optische Anreize hinzugefügt werden können. Eine Bitmap verdeckt nie das Erscheinungsbild eines Bedienelements oder Textes.

Die nachstehende Abbildung zeigt die Dialogbox *Bitmap bearbeiten*, die in der darunterstehenden Tabelle beschrieben wird.



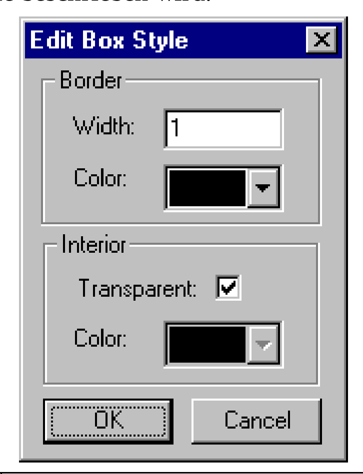
Feld	Beschreibung
Bitmap-Datei	Geben Sie Pfad und Namen der Bitmap-Datei ein oder klicken Sie auf "Durchsuchen", um eine Datei zu suchen. Es können nur Windows-Bitmapdateien (BMP) verwendet werden.

Kästchen



Ein Kästchen ist ein Anzeigeelement, mit dem Bedienelemente zusammengefaßt, Text oder Bitmaps eingerahmt, Menüteile hervorgehoben oder Menüs auf andere Weise interessant gestaltet werden können. Ein Kästchen besitzt einen Rahmen und wahlweise eine Füllfarbe. Ein Kästchen verdeckt nie das Erscheinungsbild eines Bedienelements oder Textes.

Die nachstehende Abbildung zeigt die Dialogbox *Kästchen bearbeiten*, die in der darunterstehenden Tabelle beschrieben wird.



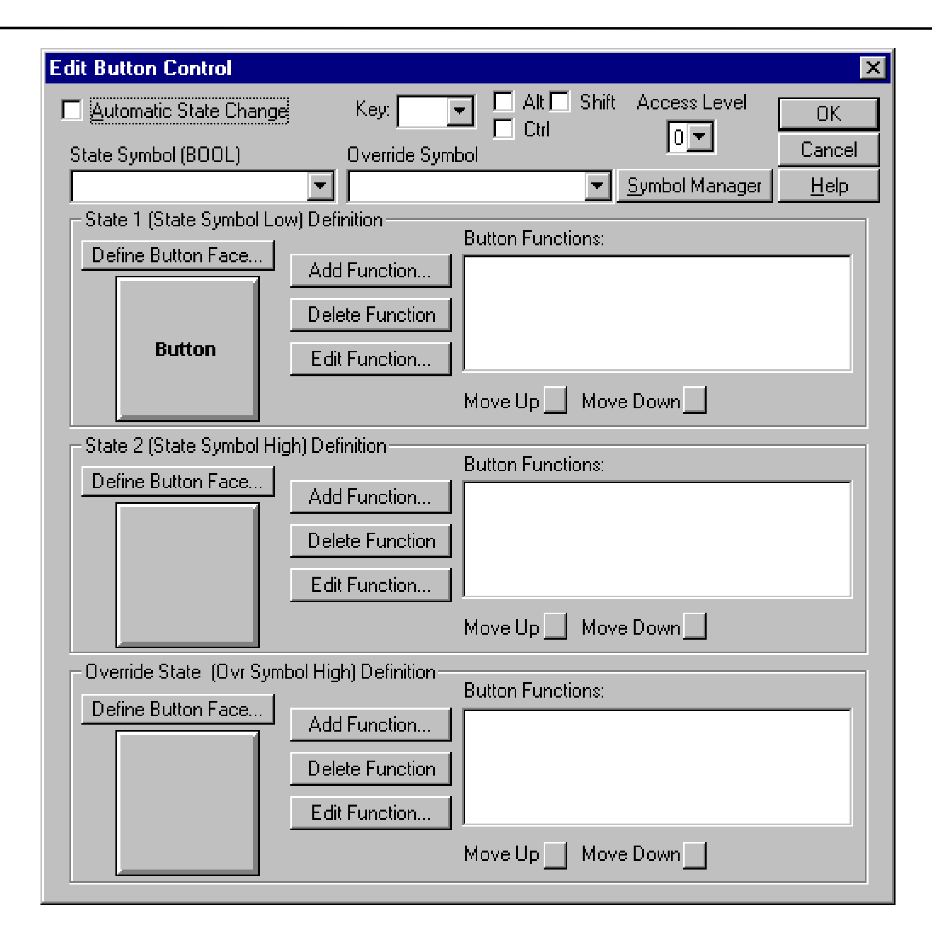
Feld	Beschreibung
Rahmen	Hiermit wird das Erscheinungsbild des Rahmens festgelegt. Geben Sie eine numerische Rahmenbreite ein und wählen Sie eine Rahmenfarbe aus dem Farben-Listenfeld.
Innen	Wird Transparent aktiviert, besitzt das Kästchen ein transparentes Inneres und die Füllfarbe ist deaktiviert. Im anderen Fall kann dem Kästchen eine Füllfarbe zugewiesen werden.

Klicktaste

Click Button

Diese Taste führt eine Funktion aus, wenn Sie auf sie klicken. Zu den Funktionen gehören Boolesches Symbol setzen oder rücksetzen, Menü anzeigen, Programm auswählen, laufen lassen, anhalten oder abbrechen, Bedienereingaben entgegennehmen, und ablauffähige Datei ausführen. Die Taste kann für Operationen mit einem Zustand oder mit zwei Zuständen mit Zustandsüberspeicherung konfiguriert werden. Für jeden Zustand können Funktionen programmiert werden. Wird auf die Taste geklickt, werden die Funktionen entsprechend dem aktuellen Tastenzustand ausgeführt. Konfiguration ist möglich für automatische Zustandsänderung oder Zustandsänderung auf der Grundlage eines Booleschen Symbols. Der Taste können eine Zugriffsebene und eine Taste auf der Tastatur (oder Kombination) zugeordnet werden.

Die nachstehende Abbildung zeigt die Dialogbox *Schaltfläche bearbeiten*, die in der Tabelle auf der nächsten Seite beschrieben wird.



Edit Button Control
Automatic State Change
Key:
Alt
Shift
Ctrl
Access Level
0
OK
Cancel
Help
State Symbol (BOOL)
Override Symbol
Symbol Manager
State 1 (State Symbol Low) Definition
Define Button Face...
Button
Add Function...
Delete Function
Edit Function...
Button Functions:
Move Up
Move Down
State 2 (State Symbol High) Definition
Define Button Face...
Add Function...
Delete Function
Edit Function...
Button Functions:
Move Up
Move Down
Override State (Ovr Symbol High) Definition
Define Button Face...
Add Function...
Delete Function
Edit Function...
Button Functions:
Move Up
Move Down

Feld	Beschreibung
Automatischer Zustandswechsel	Enthält dieses Kästchen einen Haken, dann wird der automatische Zustandswechsel aktiviert, wenn auf die Schaltfläche geklickt wird. Die Umschaltung zwischen Zustand 1 und Zustand 2 erfolgt automatisch. Bei Freigabe ist die Zustandssymbol-Editierbox gesperrt. Ist der automatische Zustandswechsel gesperrt und ist für die Schaltfläche kein Zustandssymbol definiert, dann hat die Schaltfläche nur einen Zustand und verbleibt im Zustand 1.
Zustandssymbol	Definiert das Boolesche Zustandssymbol für die Schaltfläche. Wurde für eine Schaltfläche ein Zustandssymbol definiert, dann wechselt die Schaltfläche auf Zustand 1, wenn das Zustandssymbol auf L-Pegel ist, und auf Zustand 2, wenn es auf H-Pegel ist. Geben Sie für die Schaltfläche ein Zustandssymbol ein oder wählen Sie eines aus der Liste. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*. Ist der automatische Zustandswechsel gesperrt und ist für die Schaltfläche kein Zustandssymbol definiert, dann hat die Schaltfläche nur einen Zustand und verbleibt im Zustand 1.
Symbol überspeichern	Definiert das Boolesche Überspeicherungssymbol für die Schaltfläche. Wurde für eine Schaltfläche ein Überspeicherungssymbol definiert, dann wechselt die Schaltfläche in den Überspeicherungszustand, wenn das Überspeicherungssymbol auf H-Pegel liegt. Geben Sie für die Schaltfläche ein Überspeicherungssymbol ein oder wählen Sie eines aus der Liste. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*.
Taste	Gibt eine Taste auf der Tastatur an, mit der die Schaltfläche geklickt werden kann. Sie können Funktionstasten aus einer Liste auswählen oder eine Einzeltaste im Bereich A..Z oder 1..9 eingeben. Ebenso können die Tasten Alt, Shift und Ctrl zusammen mit einer anderen Taste verwendet werden.
Zugriffsebene	Gibt eine Zugriffsebene für die Schaltfläche an. Wird eine Zugriffsebene angegeben, dann muß zum Anklicken der Schaltfläche die Zugriffsebene der Bedienerschnittstelle auf die angegebene Zugriffsebene oder höher eingestellt werden.
Schaltflächenfunktionen	Zeigt eine Liste der Steuerfunktionen für die einzelnen Zustände der Schaltfläche an. Die Steuerfunktion für den aktuellen Tastenzustand werden ausgeführt, wenn die Klicktaste gedrückt und losgelassen wird.

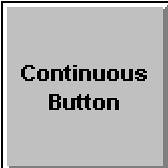
Feld	Beschreibung
Funktion hinzufügen	Fügt der Schaltflächen-Funktionsliste eine Steuerfunktion hinzu. Die Funktionen, die einer Schaltfläche zugewiesen können, sind in der Tabelle "Schaltflächenfunktionen" aufgeführt und beschrieben. Handelt es sich bei der hinzugefügten Funktion um RUN, STOP oder ABORT und wird ein Programmname angegeben, dann erscheint unterhalb der Funktionsliste eine Schaltfläche "Programm bearbeiten". Wird die Schaltfläche "Programm bearbeiten" gedrückt, dann wird der Programmeditor aktiviert und die angegebene Datei geöffnet. Existiert die Datei nicht, dann wird eine neue Datei geöffnet.
Funktion löschen	Löscht die hervorgehobene Steuerfunktion aus der Liste der Schaltflächenfunktionen.
Funktion bearbeiten	Bearbeitung der Parameter der hervorgehobenen Steuerfunktion in der Liste der Schaltflächenfunktionen.
Aufwärts	Bewegt die hervorgehobene Steuerfunktion in der Liste der Schaltflächenfunktionen um eine Stelle nach oben.
Abwärts	Bewegt die hervorgehobene Steuerfunktion in der Liste der Schaltflächenfunktionen um eine Stelle nach unten.
Oberfläche der Schaltfläche definieren	Öffnet die Dialogbox "Oberfläche der Schaltfläche definieren".
Zeile 1 Titel Zeile 2 Titel	Die Schaltfläche kann zur Beschreibung ihrer Funktion mit zwei Textzeilen versehen werden.
Schräge	Die Schräge hat Einfluß auf den Rahmen des Bedienelements. Die Werte der Schräge liegen zwischen 0 und 6. Eine Schräge von 0 läßt das Bedienelement flach erscheinen. Mit zunehmendem Wert wird die Darstellung des Bedienelements räumlicher.
Oberflächenfarbe	Gibt eine Hintergrundfarbe für das Bedienelement an.
Symbol wählen Symbol entfernen	Klicken Sie auf "Symbol wählen", um die Symbole anzuzeigen, die der Schaltfläche zugeordnet werden können.
Schriftart auswählen	Wenn Sie auf diese Schaltfläche klicken, erscheint eine Dialogbox *Schriftart*, mit der Sie Schriftart, Stil, Größe und Farbe des Bedienelementetexts auswählen können.

Schaltflächenfunktionen

Die nachstehende Tabelle führt die Steuerfunktionen auf, die den Schaltflächen-Bedienelementen zugewiesen werden können. Die Funktion wird ausgeführt, wenn Sie bei aktivem Menü auf die Schaltfläche klicken.

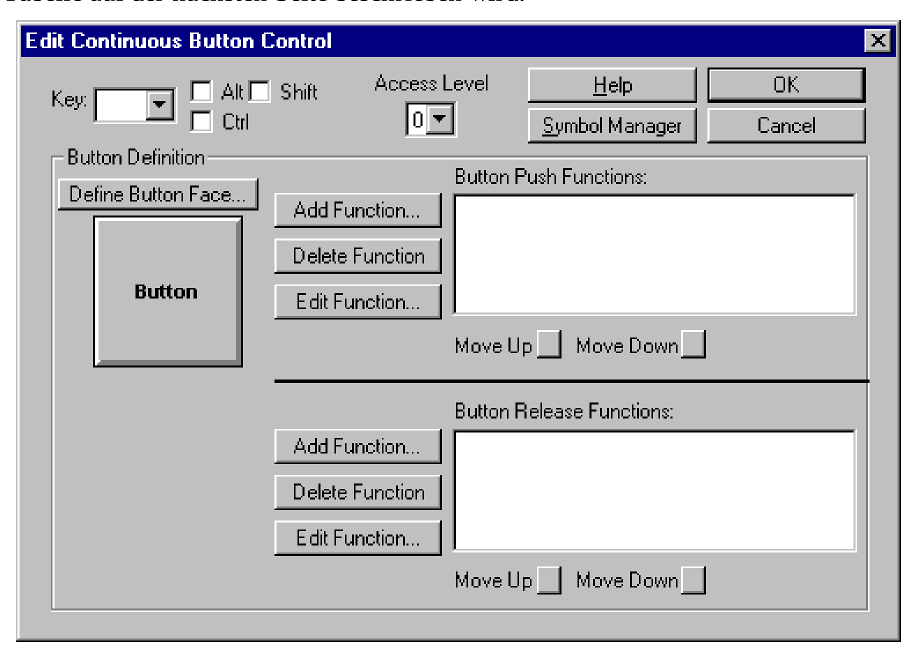
Schaltflächenfunktion	Beschreibung
SELECT PROGRAM (Programm auswählen)	Zeigt bei aktivem Menü eine Liste der AS-Programme im Projekt an, aus der der Bediener das gewünschte Programm auswählt. Die Funktionen Programm ausführen, Programm stoppen und Programm abbrechen können so eingestellt werden, daß sie auf ein vom Bediener ausgewähltes Programm (AUSGEWÄHLTES PROGRAMM) wirken.
RUN... (Programm ausführen)	Führt ein angegebenes Programm oder das vom Bediener über die Programmauswahlfunktion ausgewählte Programm (AUSGEWÄHLTES PROGRAMM) aus.
STOP... (Programm stoppen)	Stoppt eine angegebene Ablaufkette oder die vom Bediener über die Programmauswahlfunktion ausgewählte Ablaufkette (AUSGEWÄHLTES PROGRAMM). Das Programm kann mit der Funktion "Programm ausführen" von dem Punkt aus fortgesetzt werden, an dem es angehalten wurde. Es können nur Ablaufketten gestoppt werden.
ABORT... (Programm abbrechen)	Bricht ein angegebenes Programm oder das vom Bediener über die Programmauswahlfunktion ausgewählte Programm (AUSGEWÄHLTES PROGRAMM) ab.
SCREEN... (Bedienermenü aktivieren)	Aktiviert ein angegebenes Bedienermenü.
SET... (Symbol setzen)	Setzt ein angegebenes Symbol oder System-Nothalt.
CLEAR... (Symbol löschen)	Löscht ein angegebenes Symbol oder setzt System-Nothalt zurück.
OP INPUT... (Bedienereingabe)	Fragt bei der Erstellung nach Symbol und Aufforderungsmeldung. Zeigt zur Laufzeit eine Bedienereingabebox mit der angegebenen Bedieneraufforderung an. Der vom Bediener eingegebene Wert wird in dem angegebenen Symbol gespeichert.
EXECUTE... (Ablauffähige Datei ausführen)	Führt die angegebene *.exe-Datei aus. Die Ausführung eines der Produkt-Dienstprogramme erfordert die entsprechend autorisierte Zugriffsebene. Zu den Produkt-Dienstprogramme gehören Logikmanager und Programmeditor. **Hinweis:** Es kann ein Pfad angegeben werden. Enthält der Pfad Leerzeichen, dann muß die gesamte Zeichenfolge in doppelte Anführungszeichen gesetzt werden. Zum Beispiel: "C:\my .program folder\program1.exe"

Drucktaste



Eine Drucktaste führt eine Funktion aus, wenn Sie auf sie klicken. Zu den Funktionen gehören ein Boolesches Symbol setzen oder rücksetzen, ein Menü anzeigen, ein Programm auswählen, laufen lassen, anhalten oder abbrechen, Bedienereingaben entgegennehmen, und eine ablauffähige Datei ausführen. Je nach Programmierung werden die Funktionen beim Drücken oder beim Loslassen der Drucktaste ausgeführt. Der Taste können eine Zugriffsebene und eine Taste auf der Tastatur (oder Kombination) zugeordnet werden.

Die nachstehende Abbildung zeigt die Dialogbox *Drucktaste bearbeiten*, die in der Tabelle auf der nächsten Seite beschrieben wird.

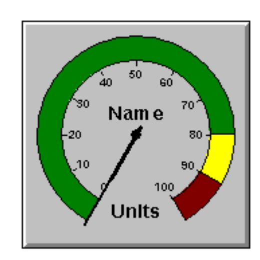


Feld	Beschreibung
Taste	Gibt eine Taste auf der Tastatur an, mit der die Schaltfläche geklickt werden kann. Sie können Funktionstasten aus einer Liste auswählen oder eine Einzeltaste im Bereich A..Z oder 1..9 eingeben. Ebenso können die Tasten Alt, Shift und Ctrl zusammen mit einer anderen Taste verwendet werden.
Zugriffsebene	Gibt eine Zugriffsebene für die Schaltfläche an. Wird eine Zugriffsebene angegeben, dann muß zum Anklicken der Schaltfläche die Zugriffsebene der Bedienerschnittstelle auf die angegebene Zugriffsebene oder höher eingestellt werden.

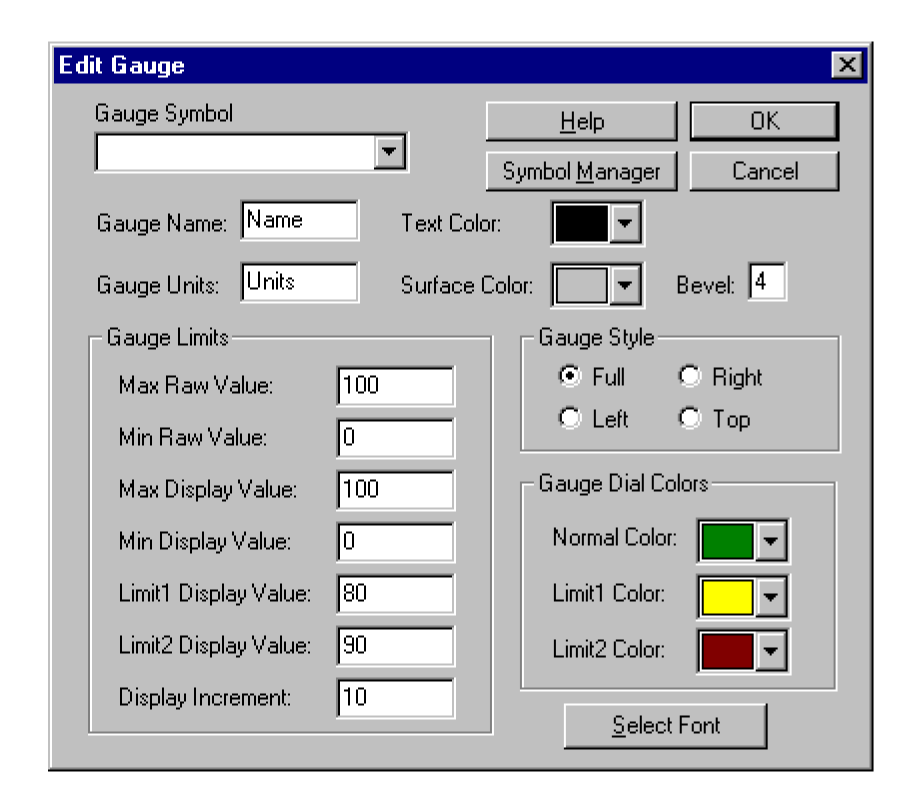
Feld	Beschreibung
Funktionen beim Drücken der Taste Funktionen beim Loslassen der Taste	Zeigt die Liste der Steuerfunktionen an, die beim Drücken der Taste ausgeführt werden. Zeigt die Liste der Steuerfunktionen an, die beim Loslassen der Taste ausgeführt werden.
Funktion hinzufügen	Fügt einer der beiden vorgenannten Listen eine Steuerfunktion hinzu. Die Funktionen, die einer Schaltfläche zugewiesen können, sind in der Tabelle "Schaltflächenfunktionen" aufgeführt und beschrieben. Handelt es sich bei der hinzugefügten Funktion um RUN, STOP oder ABORT und wird ein Programmname angegeben, dann erscheint unterhalb der Funktionsliste eine Schaltfläche "Programm bearbeiten". Wird die Schaltfläche "Programm bearbeiten" gedrückt, dann wird der Programmeditor aktiviert und die angegebene Datei geöffnet. Existiert die Datei nicht, dann wird eine neue Datei geöffnet.
Löschfunktion	Löscht die selektierte Steuerfunktion aus einer der beiden vorgenannten Listen.
Editierfunktion	Zur Bearbeiten der Parameter aus der selektierten Steuerfunktion in einer der beiden vorgenannten Listen.
Aufwärts	Bewegt die selektierte Steuerfunktion in einer der beiden vorgenannten Listen um eine Position nach oben.
Abwärts	Bewegt die selektierte Steuerfunktion in einer der beiden vorgenannten Listen um eine Position nach unten.
Oberfläche der Schaltfläche definieren	Öffnet die Dialogbox "Oberfläche der Schaltfläche definieren".
Zeile 1 Titel Zeile 2 Titel	Die Schaltfläche kann zur Beschreibung ihrer Funktion mit zwei Textzeilen versehen werden.
Schräge	Die Schräge hat Einfluß auf den Rahmen des Bedienelements. Die Werte der Schräge liegen zwischen 0 und 6. Eine Schräge von 0 läßt das Bedienelement flach erscheinen. Mit zunehmendem Wert wird die Darstellung des Bedienelements räumlicher.
Oberflächenfarbe	Gibt eine Hintergrundfarbe für das Bedienelement ein.
Symbol wählen Symbol entfernen	Klicken Sie auf "Symbol wählen", um die Symbole anzuzeigen, die der Schaltfläche zugeordnet werden können.
Schriftart auswählen	Wenn Sie auf diese Schaltfläche klicken, erscheint eine Dialogbox *Schriftart*, mit der Sie Schriftart, Stil, Größe und Farbe des Bedienelementetexts auswählen können.

Anzeigeuhr

Die Anzeigeuhr besitzt eine Skala und einen drehbaren Zeiger. Der Drehzeiger gibt den aktuellen Wert des der Uhr zugeordneten Symbols an. Ändert sich der Symbolwert, bewegt sich der Zeiger im oder entgegen dem Uhrzeigersinn, um den aktuellen Symbolwert anzuzeigen. Sie können Grenzwerte eingeben, durch die die Skalenfarbe entsprechend einer Vorgabe verändert wird. Es gibt vier verschiedene Uhrtypen.



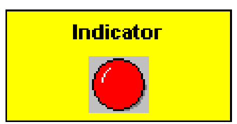
Die nachstehende Abbildung zeigt die Dialogbox *Anzeigeuhr bearbeiten*, die in der Tabelle darunter beschrieben wird.



Feld	Beschreibung
Anzeigeinstrument	Spezifiziert das Symbol, dessen Wert durch den Drehzeiger angezeigt wird. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*.

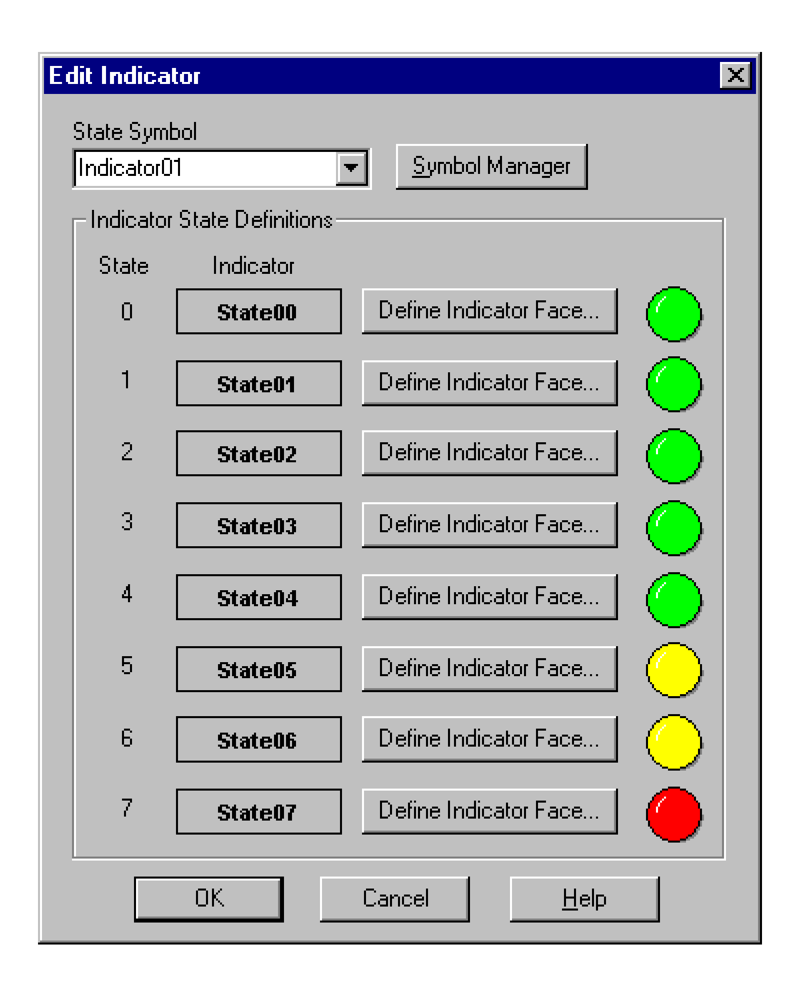
Feld	Beschreibung
Instrumentenname	Legt für das Bedienelement einen Namen fest. Nur zur Menüdokumentation.
Anzeigeeinheiten	Legt für das Bedienelement eine Einheit fest. Nur zur Menüdokumentation.
Textfarbe	Legt eine Farbe für Name, Einheiten und Skalennumerierung des Bedienelements fest.
Oberflächenfarbe	Gibt eine Hintergrundfarbe für das Bedienelement ein.
Schräge	Die Schräge hat Einfluß auf den Rahmen des Bedienelements. Die Werte der Schräge liegen zwischen 0 und 6. Eine Schräge von 0 läßt das Bedienelement flach erscheinen. Mit zunehmendem Wert wird die Darstellung des Bedienelements räumlicher.
Anzeige-Grenzwerte	
Max. Rohwert Min. Rohwert	Dies sind die oberen und unteren Grenzwerte, auf die der Zeiger des Anzeigeinstruments anspricht. Sie begrenzen den Wert des vom Zeiger dargestellten Symbols. Alle Werte, die größer als der maximale Rohwert sind, setzen den Zeiger auf seine Obergrenze. Alle Symbolwerte, die unterhalb des minimalen Rohwertes liegen, setzen den Zeiger auf seine Untergrenze. In der Standardeinstellung bewegt sich der Zeiger bei steigenden Symbolwerten in Uhrzeigerrichtung. Um den Zeiger im Gegenuhrzeigersinn zu bewegen müssen Sie die maximalen und minimalen Rohwerte vertauschen.
Max. Anzeigewert Min. Anzeigewert	Gibt die maximale und minimale Skalennumerierung an.
Grenze1 Anzeigewert Grenze2 Anzeigewert	Gibt die Grenzwerte an, bei denen die Skala des Anzeigeinstruments ihre Farbe wechselt. Zum Beispiel zur Anzeige von Alarmzuständen.
Anzeigeinkrement	Gibt das Skalennumerierungsinkrement an.
Instrumentenart	Vollkreis. Halbkreis mit Bogen nach oben. Halbkreis mit Bogen nach links. Halbkreis mit Bogen nach rechts.
Farben der Instrumentenskala	Normal - die Farbe der Anzeigenskala unterhalb eines Grenzwertes. Grenze 1 - die Farbe der Anzeigenskala, wenn die angegebene Grenze 1 erreicht ist. Grenze 2 - die Farbe der Anzeigenskala, wenn die angegebene Grenze 2 erreicht ist.
Symbolmanager	Klicken Sie auf diese Schaltfläche, um den Symbolmanager zu öffnen.
Schriftart auswählen	Wenn Sie auf diese Schaltfläche klicken, erscheint eine Dialogbox *Schriftart*, mit der Sie Schriftart, Stil, Größe und Farbe des Bedienelementetexts (Name, Einheiten und Skalennumerierung) auswählen können.

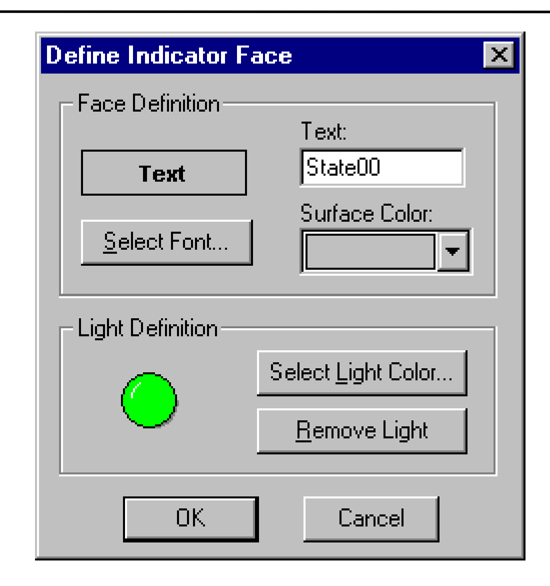
Kennmelder



Ein Kennmelder ist eine Zustandssteuerung, deren Erscheinungsbild (Text, Farbe, usw.) sich entsprechend dem Wert eines Symbols vom Typ BOOL oder BYTE verändert. Bei einem BOOL-Symbol sind zwei unterschiedliche Zustände möglich, bei einem BYTE-Symbol sind es acht unterschiedliche Zustände.

Die nachstehende Abbildung zeigt die Dialogboxen *Kennmelder bearbeiten* und *Kennmelder-Aussehen definieren* die in der Tabelle auf der nächsten Seite beschrieben werden.





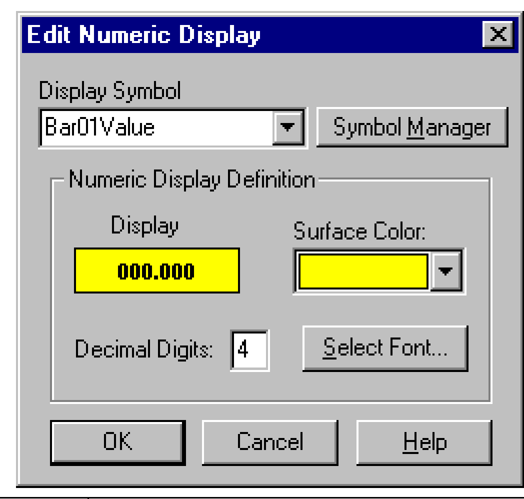
Feld	Beschreibung
Zustandssymbol	Geben Sie das Symbol an, auf das der Kennmelder reagiert. Die Zustandssymbol-Datentypen können BOOL oder BYTE sein. Ist das Zustandssymbol vom Typ BOOL; kann der Kennmelder zwei Zustände annehmen. Ist es vom Typ BYTE, sind acht verschiedene Zustände möglich. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*.
Symbolmanager	Klicken Sie auf diese Schaltfläche, um den Symbolmanager zu öffnen.
Kennmelder-Zustandsdefinitionen	Für jeden möglichen Binärzustand (0 bis 7) werden die entsprechenden Möglichkeiten für Anzeigetext, Hintergrundfarbe und Zusatzlicht angezeigt.
Kennmelder-Aussehen definieren	Klicken Sie auf diese Schaltfläche, um die Dialogbox *Kennmelder-Aussehen definieren* aufzurufen. Mit dieser Dialogbox werden Text,Textfarbe und Hintergrundfarbe eines Zustands sowie Schriftart und -größe für den gesamten Kennmelder definiert. Für jeden Kennmelderzustand kann zusätzlich eine Farbleuchte definiert werden.
Frontdefinition Text Schriftart auswählen Oberflächenfarbe	 Die Zeichenfolge, die für den Zustand angezeigt werden soll. Definiert Schriftart und -größe für Kennmelder und Textfarbe für Zustand. Definiert die Hintergrundfarbe für den Zustand.
Leuchtendefinition Leuchte wählen Leuchte entfernen	 Zeigt eine Gruppe von Leuchtenfarben an, die für den Zustand ausgewählt werden können. Entfernt eine für den Zustand definierte Leuchte.

Zahlenanzeige

Mit der numerischen Anzeige wird der Wert eines Symbols (numerisch oder Zeichenfolge) angezeigt.



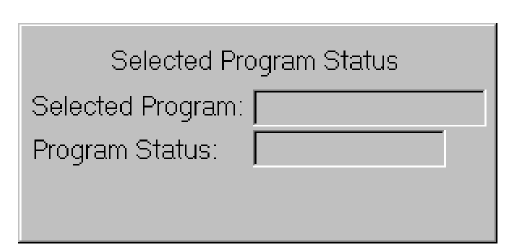
Die nachstehende Abbildung zeigt die Dialogbox *Zahlenanzeige bearbeiten*, die in der Tabelle auf der nächsten Seite beschrieben wird.



Feld	Beschreibung
Anzeigesymbol	Spezifiziert das Symbol, dessen Wert in der numerischen Anzeige erscheint. Wählen Sie einen Symbolnamen aus dem Drop-Down-Listenfeld oder geben Sie einen Symbolname in das Editierfeld ein. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*.
Symbolmanager	Klicken Sie auf diese Schaltfläche, um den Symbolmanager zu öffnen.
Anzeige	Die ausgewählte Text- und Hintergrundfarbe sowie der Schrifttyp werden in dem Kästchen *Anzeige* dargestellt.
Oberflächenfarbe	Definiert die Hintergrundfarbe für das Bedienelement.
Dezimalstellen	Definiert die Anzahl Stellen, die zur Rechten des Dezimalpunktes angezeigt werden.
Schriftart auswählen	Definiert Schriftare, Schriftgröße und Schriftfarbe des Bedienelementes.

Auswahlprogramm-Statustafel

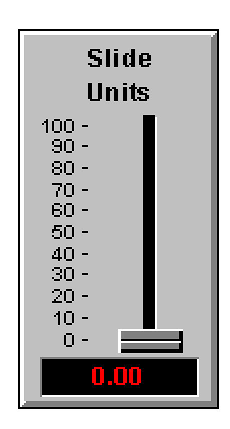
Die Auswahlprogramm-Statustafel zeigt Namen und aktuellen Status (z.B. läuft, gestoppt, usw.) des gewählten Programms an. Das Auswahlprogramm ist das Programm, das Sie über eine Klicktaste oder Drucktaste mit dem Befehl PROGRAMM AUSWÄHLEN ausgewählt haben.



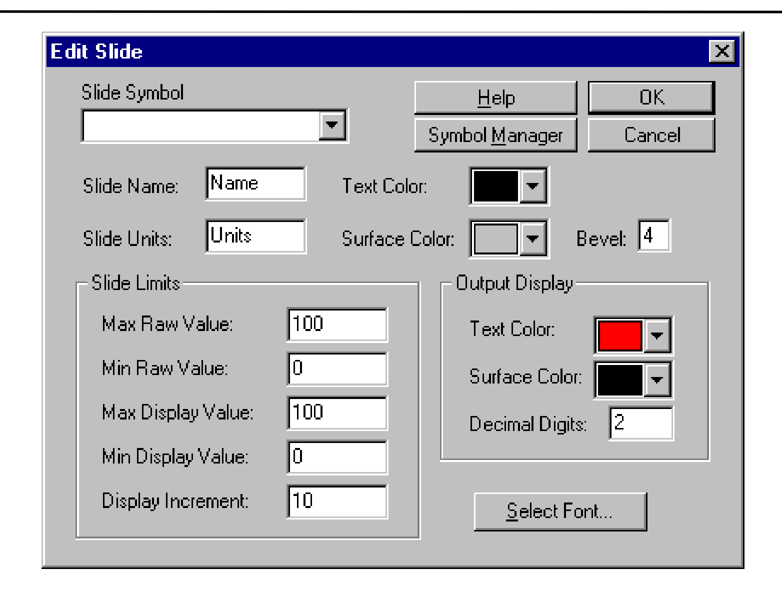
Die Auswahlprogramm-Statustafel besitzt keine Konfigurationsoptionen.

Schieber

Mit dem Schieber können Sie einen Symbolwert kontinuierlich verändern. Wird der Schieber nach oben oder unten bewegt, dann verändert sich der Wert des dem Schieber zugeordneten Symbols. Die aktuelle Einstellung des Schiebers wird unten am Bedienelement angezeigt. Eine Skala entlang dem Schieberweg zeigt ungefähr den eingestellten Wert an.



Die nachstehende Abbildung zeigt die Dialogbox *Schieber bearbeiten*, die in der Tabelle auf der nächsten Seite beschrieben wird.



Feld	Beschreibung
Schiebersymbol	Spezifiziert das Symbol, dessen Wert durch den Schieber verändert wird. Falls erforderlich, kann ein neues Symbol erzeugt werden. Öffnen Sie hierzu den Symbolmanager mit der Schaltfläche *Symbolmanager*.
Schiebernamen	Legt für das Bedienelement einen Namen fest. Nur zur Menüdokumentation.
Schiebereinheiten	Legt für das Bedienelement eine Einheit fest. Nur zur Menüdokumentation.
Textfarbe	Legt eine Farbe für Name, Einheiten und Skalennumerierung des Bedienelements fest.
Oberflächenfarbe	Gibt eine Hintergrundfarbe für das Bedienelement ein.
Schräge	Die Werte der Schräge liegen zwischen 0 und 6. Eine Schräge von 0 läßt das Bedienelement flach erscheinen. Mit zunehmendem Wert wird die Darstellung des Bedienelements räumlicher.
Schieber-Grenzwerte	Die Rohwerte legen den Zusammenhang zwischen Bedienelement und Symbol fest. Die Anzeigewerte bestimmen den Zusammenhang zwischen Bedienelement und Skala.
Max. Rohwert Min. Rohwert	Gibt den minimalen und den maximalen Wert an, den der Schiebe dem Symbol zuweisen kann. Der Schieber setzt keine Werte oberhalb des Maximalwertes oder unterhalb des Minimalwertes.
Max. Anzeigewert Min. Anzeigewert	Gibt die maximale und minimale Skalennumerierung an.
Anzeigeinkrement	Gibt das Skalennumerierungsinkrement an.

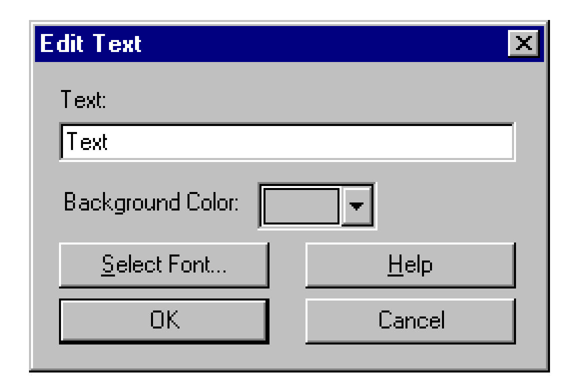
Feld	Beschreibung
Ausgabeanzeige	Die Ausgabeanzeige zeigt die Bedienelemente-Anzeige an (nicht den Symbolwert). D.h. die Position des Schiebers bezogen auf die Skala.
Textfarbe	Definiert die Textfarbe der Ausgabeanzeige.
Oberflächenfarbe	Definiert die Hintergrundfarbe der Ausgabeanzeige.
Dezimalstellen	Definiert die Anzahl Stellen, die zur Rechten des Dezimalpunktes angezeigt werden.
Symbolmanager	Klicken Sie auf diese Schaltfläche, um den Symbolmanager zu öffnen.
Schriftart auswählen	Wenn Sie auf diese Schaltfläche klicken, erscheint eine Dialogbox *Schriftart*, mit der Sie Schriftart, Stil, Größe und Farbe des Bedienelementetexts (Name, Einheiten und Skalennumerierung) auswählen können.

Text

Mit dem Textelement können Sie Bedienelemente und Bedienelementgruppen beschriften, Menüfunktionen dokumentieren, oder sonstige Beschreibungen von Menüoperationen oder Menüfunktionen erzeugen.



Die nachstehende Abbildung zeigt die Dialogbox *Text bearbeiten*, die in der darunterstehenden Tabelle beschrieben wird.



Feld	Beschreibung
Text	Geben Sie den anzuzeigenden Text in das Textfeld ein.
Hintergrundfarbe	Definiert die Hintergrundfarbe für das Textelement.
Schriftart auswählen	Definiert Schriftart, Schriftgröße und Schriftfarbe der Zeichenfolge.

ActiveX-Bedienelemente

Einleitung

ActiveX-Bedienelemente sind DLLs mit einer OLE2-Schnittstelle, die standardmäßig eine OCX-Dateinnamenerweiterung verwenden (anstelle von DLL). Zu den ActiveX-Bedienelementen gehören einfache Bedienelemente, wie Schaltflächen oder Editierboxen, komplexere Bedienelemente wie Anzeigeinstrumente sowie DLL-Komponenten, die keine Bedienerschnittstelle besitzen. ActiveX-Bedienelemente benötigen einen Container, der von der Bedienerschnittstelle bereitgestellt wird.

ActiveX-Bedienelemente besitzen drei Funktionstypen:

- **Eigenschaften** – Diese Funktionen innerhalb des Bedienelements können vom Container zur Konfiguration des Erscheinungsbildes des Bedienelements aufgerufen werden (z.B. Farben, Schriftarten, Bildunterschriften, Bitmaps, usw.).
- **Methoden** – Diese Funktionen innerhalb des Bedienelements gestatten dem Container, Daten über das Bedienelement zu senden oder zu empfangen. Sie können zum Beispiel mit einer Methode den Wert einer Zelle in einem Gitternetzsteuerung zu holen oder die Steuerung anzuweisen, sich selbst umzufärben.
- **Ereignisse –** Diese Funktionen werden aufgerufen, wenn innerhalb des Bedienelements ein Windows-Ereignis (z.B. ein Mausklick oder ein Tastendruck) auftritt. Der Container wird vom Auftreten eines dieser Ereignisse benachrichtigt.

Eigenschaften, Methoden und Ereignisse werden mit dem Bedienerschnittstellenmenü-Editor bearbeitet, nachdem das Bedienelement in einem Menü hinzugefügt wurde.

ActiveX-Einschränkungen

Die nachstehende Liste gibt die Einschränkungen bei der Benutzung der ActiveX-Bedienelemente in der Bedienerschnittstelle an:

1. Keine Unterstützung von Verbund-Bedienelementen (Bedienelemente innerhalb anderer Bedienelemente).
2. Kein Typenzwang (von DWORD auf INT, zum Beispiel).
3. Keine Unterstützung von Bedienelementen, die keine eigene Eigenschaftenseite besitzen.

4. Keine Unterstützung beim Verändern von Schriftaren während der Laufzeit. Die Bedienerschnittstelle unterstützt keinen Typ "Schriftart".
5. Keine Unterstützung von Methoden-/Ereignisparametern vom Typ VT_VARIANT.
6. Keine Unterstützung von komplexen Datentypen (Strukturen) für Methoden-/Ereignisparameter.

ActiveX- und Standard-Bedienelemente

Die mit dem Produkt gelieferten Standard-Bedienelemente vermitteln die Grundfunktionalität für eine programmierbare Steuerungssystem-Bedienerschnittstelle. Diese Bedienelemente besitzen auch interne Kenntnisse des programmierbaren Steuerungssystems. Diese Kenntnisse ermöglichen es, zahlreiche komplexe Einzelheiten dem Endbenutzer gegenüber auszublenden.

Zur Erweiterung der Funktionalität bzw. zur Erfüllung spezieller Forderungen dient die Bedienerschnittstelle als Container für ActiveX-Bedienelemente. Alle verfügbaren ActiveX-Bedienelemente können in einem Bedienerschnittstellenmenü abgelegt werden und Daten mit der Runtimemaschine austauschen (sofern die Datentypen mit der Steuerungssystemsoftware kompatibel sind).

Beachten Sie folgende Punkte:

- Entsprechend erfahrene Anwender können Ihre eigenen ActiveX-Bedienelemente schreiben oder von anderen Quellen übernehmen. Es liegt in Ihrer Verantwortung, die Eignung eines von Dritten übernommenen ActiveX-Bedienelements zu überprüfen und die zugehörige Dokumentation zu erhalten. Schlecht funktionierende Bedienelemente können Probleme verursachen. Nehmen Sie mit den Hersteller des Bedienelements Kontakt auf, wenn sich aus der Benutzung eines ActiveX-Bedienelements heraus Probleme ergeben.
- ActiveX-Bedienelements von anderen Herstellern besitzen **keine** internen Kenntnisse des programmierbaren Steuerungssystems. Zu den **nicht** vorhandenen Kenntnissen gehören zum Beispiel: Wie wird ein NOTHALT befohlen und wie werden die Programme des programmierbaren Steuerungssystems gestartet, angehalten und abgebrochen.
- Die ActiveX-Bedienelemente benötigen eine Registrierung, damit sie von Windows NT oder Windows 95 erkannt werden. Es ist möglich, daß der Hersteller des Bedienelements für die Registrierung der Bedienelemente sorgt. Für die Registrierung der Bedienelemente enthält der Bedienerschnittstellenmenü-Editor einen Registrierungsprozeß.

Quellen für ActiveX-Bedienelemente

Es wurden zwei ActiveX-Bedienelementepakete anderer Hersteller getestet, die als Quelle für ActiveX-Bedienelemente empfohlen werden:

VICOMponents™ Version 4.0 von ComputerBoards, Inc.
ComputerBoards Inc.
125 High Street
Mansfield, MA 02048
www.computerboards.com

Global Majic Software ActiveX controls.
Global Majic Software, Inc.
P.O. Box 322
Madison, Alabama 35758
gms@globalmajic.com
www.globalmajic.com

Registrierung von ActiveX-Bedienelementen

Ein ActiveX-Bedienelement muß registriert werden, ehe es in einem Bedienerschnittstellenmenü verwendet werden kann. Wurde ein ActiveX-Bedienelement nicht zuvor registriert, dann kann es im Bedienerschnittstellenmenü-Editor registriert werden. Registrierte Bedienelemente erscheinen in der Liste, wenn Sie im Menü *Bearbeiten* die Option *ActiveX-Bedienelement einfügen* wählen.

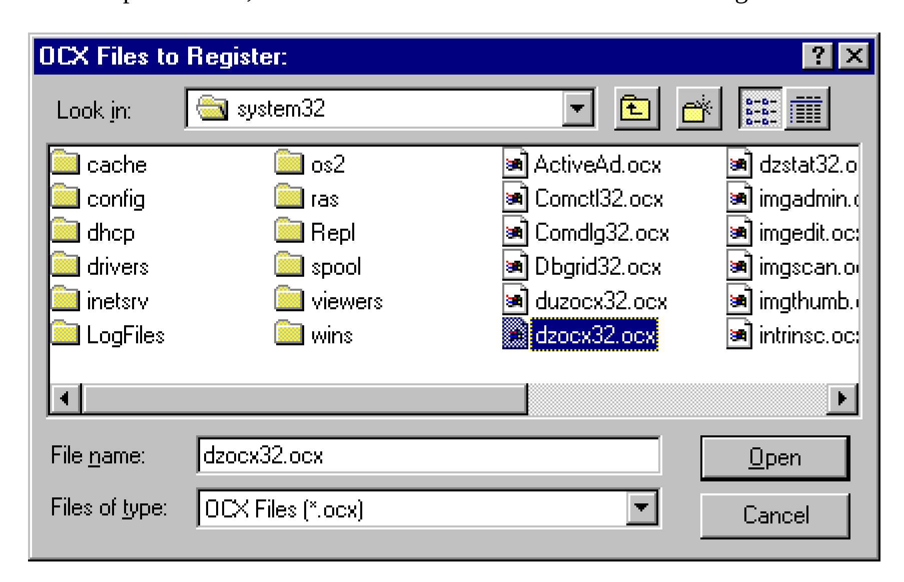
Registrierung eines ActiveX-Bedienelements

1. Wählen Sie im Menü *Bearbeiten* die Option *ActiveX-Bedienelement registrieren*. Hierauf erscheint die Dialogbox *OCX-Dateien zum Registrieren*.

2. Selektieren Sie das Bedienelement, das Sie registrieren wollen, und klicken dann auf *Öffnen*, um es zu registrieren. War die Registrierung erfolgreich, wird dies durch eine entsprechende Meldung bestätigt.

Hinweis

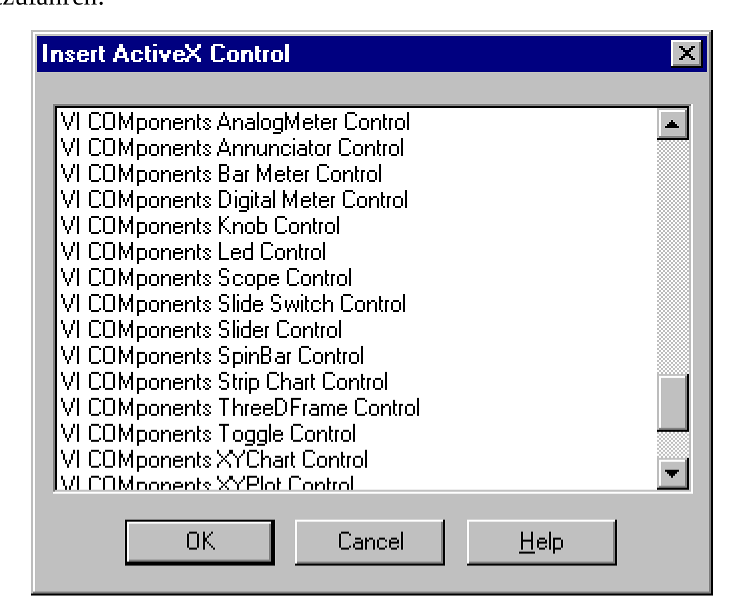
Dieser Vorgang gewährleistet nicht, daß das Bedienelement richtig in ein Menü eingefügt werden kann. Wurde das Bedienelement zum Beispiel nicht lizenziert oder nicht richtig implementiert, dann können Sie es nicht in ein Menü einfügen.



Einfügen von ActiveX-Bedienelementen

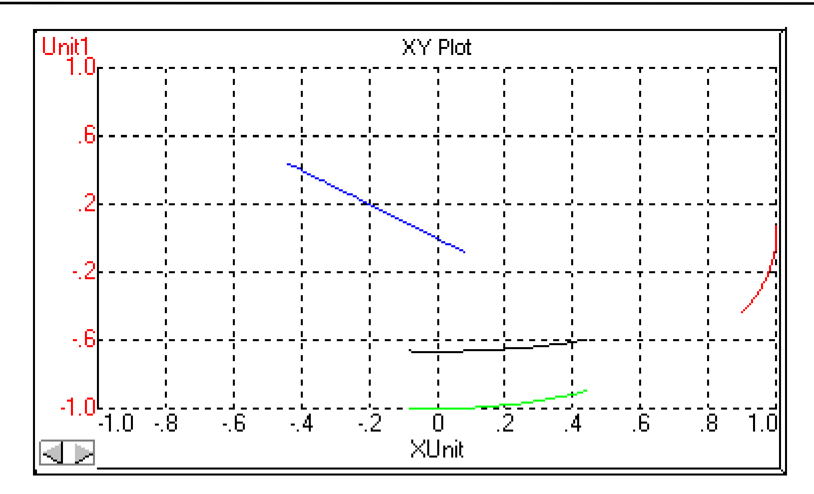
Einfügen eines ActiveX-Bedienelements

1. Wählen Sie im Menü *Bearbeiten* die Option *ActiveX-Bedienelement einfügen*. Hierauf erscheint die Dialogbox *ActiveX-Bedienelement einfügen*.
2. Wählen Sie ein ActiveX-Bedienelement und klicken dann auf *OK*, um fortzufahren.



Alternativ hierzu erscheint unten im Menü *Bearbeiten* eine Liste der letzten vier eingefügten ActiveX-Bedienelemente. Hieraus können Sie das Bedienelement auswählen, das Sie einfügen wollen.

Die nachstehende Abbildung zeigt ein Beispiel des Bedienelements XY Plot von VICOMponents.



Hinweis

Die Liste enthält alle auf dem PC registrierten ActiveX-Bedienelemente. Bei einigen ist der Einsatz in einer Steuerungsumgebung wenig sinnvoll.

Wenn Sie eine Bedienerschnittstellenmenükonfiguration mit einem ActiveX-Bedienelement verteilt anordnen, dann muß das ActiveX-Bedienelement im Zielsystem registriert sein.

Fragen Sie den Hersteller des Bedienelements über Verwendung und Verteilungsrechten.

Bearbeitung von ActiveX-Bedienelementen

ActiveX-Bedienelemente können wie Standard-Bedienelemente verschoben und in der Größe verändert werden. Die Wirkung einer Größenveränderung hängt von dem Bedienelement ab und ergibt nicht immer das gewünschte Ergebnis. Die anderen Bearbeitungsvorgänge (wie Löschen, Ausschneiden, Einfügen oder Kopieren) funktionieren wie bei Standard-Bedienelementen.

Eigenschaften bearbeiten

Führen Sie eine der folgenden Aktionen zum Bearbeiten der Eigenschaften durch:

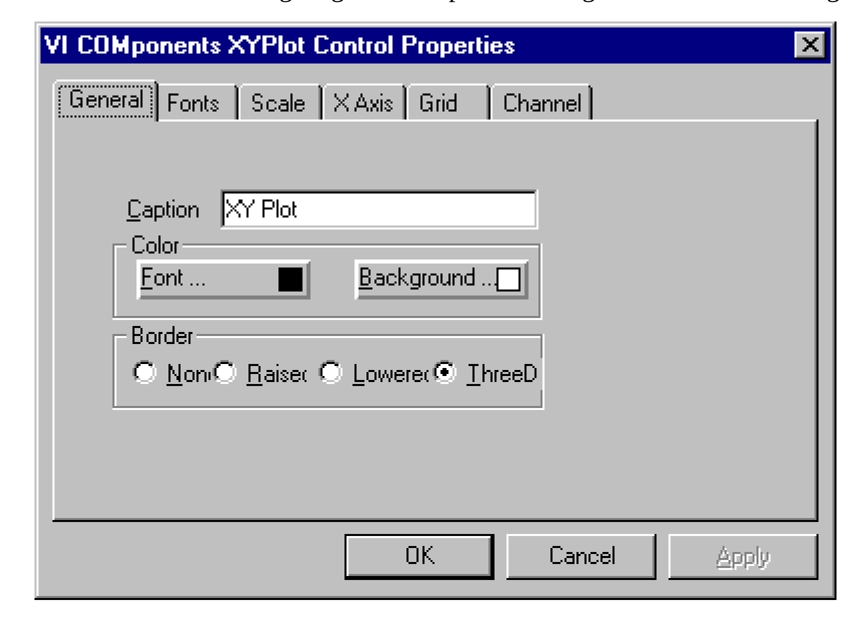
- Selektieren Sie das ActiveX-Bedienelement und wählen dann *ActiveX-Eigenschaften bearbeiten* aus dem Menü *Bearbeiten* oder dem Kontextmenü, oder benutzen Sie die Symbolleiste.
- Doppelklicken Sie auf das ActiveX-Bedienelement und wählen Sie dann *Eigenschaftsseiten bearbeiten* aus der nun erscheinenden Dialogbox *ActiveX-Bearbeitungsmodus auswählen*.

Die Eigenschaftsseitendialoge kommen aus dem Bedienelement heraus. Dies bedeutet, daß die Eigenschaftsseiten-Dialogboxen der einzelnen Bedienelemente unterschiedlich sind. Bei Bedienelementen, die keine Eigenschaftsseiten-Dialogbox besitzen, erscheint stattdessen eine entsprechende Meldung.

Hinweis

Informationen zu den Bedienelemente-Eigenschaften erhalten Sie vom Lieferanten des ActiveX-Bedienelements. Diese Informationen sind im Bedienerschnittstellenmenü-Editor selbst nicht verfügbar. Wird mit dem Bedienelement jedoch eine Hilfedatei geliefert, kann diese von der Dialogbox aus aufgerufen werden.

Die nachstehende Abbildung zeigt das Beispiel einer Eigenschaftsseiten-Dialogbox.



Methoden bearbeiten

Zur Konfiguration von Methoden der ActiveX-Bedienelemente sind drei allgemeine Schritte erforderlich:

- Wählen Sie aus, welche der Methoden des Bedienelements von der Bedienerschnittstelle aufgerufen wird.
- Weisen Sie den Argumenten der Methode Symbole und einen Rückgabewert (falls erforderlich) zu.
- Definieren Sie, wann die Methode aufgerufen wird.

Bedienelemente besitzen eine unterschiedliche Anzahl interner Methoden, wobei jede Methode ein Funktionsaufruf zum Bedienelement ist. Die Argumente und der Rückgabewert einer Funktion können aus einem von vielen möglichen (derzeit 38) Datentypen sein. Es ist nicht immer möglich, daß einer dieser Datentypen auf einem vom System unterstützten Datentyp abgebildet werden kann. Ist es nicht möglich, einen unterstützten Datentyp auf einen Typ abzubilden, der in der Methode erwartet wird, dann kann diese Methode nicht aufgerufen werden. Bei einem Versuch, diese Methode einzurichten, erscheint eine entsprechende Fehlermeldung.

Die Argumenttypen für ActiveX-Bedienelemente beginnen mit *VT_*. Die folgende Tabelle zeigt, wie die Typen auf die unterstützten Datentypen abgebildet werden:

Datentyp des ActiveX-Bedienelements	Vom System unterstützter Datentyp
VT_BOOL	BOOL
VT_UI1, VT_I1	BYTE
VT_UI2	WORD
VT_UI4, VT_UINT	DWORD
VT_I2, VT_INT	INT
VT_I4	DINT
VT_R4, VT_R8	REAL
VT_BSTR	STRING
VT_VOID, VT_EMPTY, VT_NULL	nicht abgebildet
VT_PTR	Pointer auf alle vorstehenden VT_ Typen werden unterstützt.

Ein Teil der Konfiguration eines ActiveX-Bedienelementes liegt darin zu definieren, wann die Methode des Bedienelements aufgerufen werden soll. Dies erfolgt auf der Grundlage eines 200-Millisekunden-Timers, der in der Bedienerschnittstelle abläuft. Der Aufruf einer Methode kann alle 200 Millisekunden oder einem Vielfachen von 200 Millisekunden erfolgen. Das Vielfache kann anwenderspezifisch bis zu einem Wert von 100 (= 20 Sekunden) eingestellt werden.

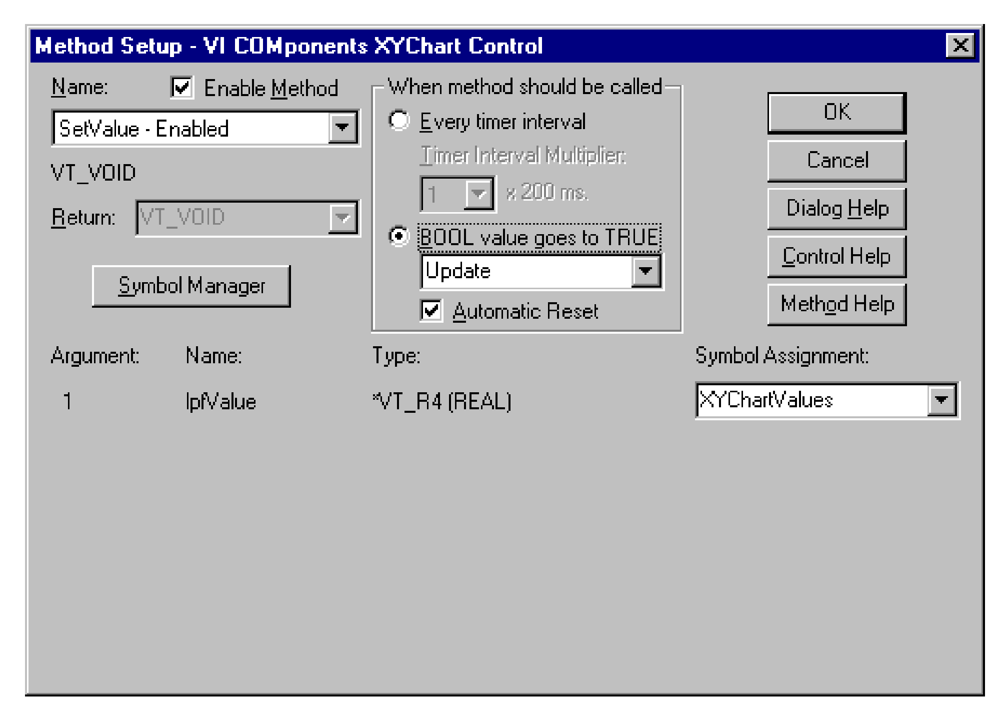
Eine Methode kann auf zwei verschiedene Arten aufgerufen werden:

In jedem Timerintervall – oder einem anwenderdefiniert Vielfachen des Timerintervalls.

Wenn ein Boolesches Symbol TRUE wird – wird ein anwenderdefiniertes Boolesches Symbol TRUE, dann wird die Methode im nächsten Timerintervall aufgerufen. Bei Rückkehr von der Methode wird das Boolesche Symbol wahlweise auf FALSE gesetzt.

Methoden bearbeiten

1. Führen Sie einen der folgenden Schritte durch:
   - Selektieren Sie das ActiveX-Bedienelement und wählen dann *ActiveX-Methoden bearbeiten* aus dem Menü *Bearbeiten* oder dem Kontextmenü, oder benutzen Sie die Symbolleiste.
   - Doppelklicken Sie auf das ActiveX-Bedienelement und wählen Sie dann *Methoden bearbeiten* aus der nun erscheinenden Dialogbox *ActiveX-Bearbeitungsmodus auswählen*.
2. Es erscheint eine Dialogbox *Methodeneinstellung*. Nehmen Sie die nachstehende Tabelle zur Hilfe und konfigurieren Sie die Methoden entsprechend den Anforderungen. Am Ende der Arbeiten klicken Sie auf *OK*, um die Konfiguration zu speichern und zum Bedienerschnittstellenmenü-Editor zurückzukehren.



Feld	Beschreibung
Name	Wählen Sie die zu konfigurierende Methode aus dem Namen-Listenfeld. Die Methoden erscheinen in alphabetischer Reihenfolge.
Methode freigeben	Aktivieren Sie dieses Kontrollkästchen, um die Methode freizugeben. Die Methode kann nur dann aufgerufen werden, wenn sie freigegeben wurde.
Rückkehr	Besitzt die Methode einen Rückgabetyp VOID, dann erscheint VT_VOID im Kästchen Rückgabe. Dieser Wert kann nicht verändert werden. In andern Fall wird der Rückgabetyp angegeben und es muß im Rückgabe-Listenfeld ein Symbol des entsprechenden Typs zugewiesen werden. Falls erforderlich, klicken Sie auf *Symbolmanager*, um ein neues Symbol zu erzeugen.

Feld	Beschreibung
Argumente	Akzeptiert die Methode Argumente, dann werden diese mit Nummer, Namen und Datentyp aufgelistet. Die Methoden sind auf 16 Argumente beschränkt. Eine Fehlermeldung erscheint, wenn die Methode mehr als 16 Argumente umfaßt. Für jedes Argument, dem ein Wert zugewiesen werden muß, weisen Sie ein Symbol aus seiner Symbolzuweisungsliste zu. Falls erforderlich, klicken Sie auf *Symbolmanager*, um neue Symbole zu erzeugen. Pointer auf Felder (definiert im Symbolmanager) können an die Methode weitergegeben werden. Geben Sie den Feldnamen ohne Index an, um das gesamte Feld weiterzugeben. Die Größe des Feldes muß der für die Methode geforderten Größe entsprechen.
Aufruf der Methode	Soll die Methode nur entsprechend einem Zeitintervall aufgerufen werden, klicken Sie auf *Jedes Timerintervall* und stellen Sie einen Multiplikator ein. Soll die Methode aufgerufen werden, wenn ein Boolesches Symbol TRUE wird, dann wählen Sie einen Zeitintervall-Multiplikator, klicken anschließend auf *Boolescher Wert wird TRUE* und wählen dann ein Boolesches Symbol aus dem Listenfeld. Wurde das Kontrollkästchen *Automatisches Rücksetzen* aktiviert, wird das Boolesche Symbol auf FALSE rückgesetzt, nachdem die Methode aufgerufen wurde.
Symbolmanager	Zeigt den Symbolmanager an.
Dialoghilfe	Zeigt Hilfe zur Methodeneinstellungs-Dialogbox an.
Bedienelement-Hilfe	Zeigt Hilfe zum ActiveX-Bedienelement an (vom Bedienelemente-Hersteller geliefert).
Methoden-Hilfe	Zeigt Hilfe zur aktuell angezeigten ActiveX-Bedienelementenmethode an (vom Bedienelemente-Hersteller geliefert).

Ereignisse bearbeiten

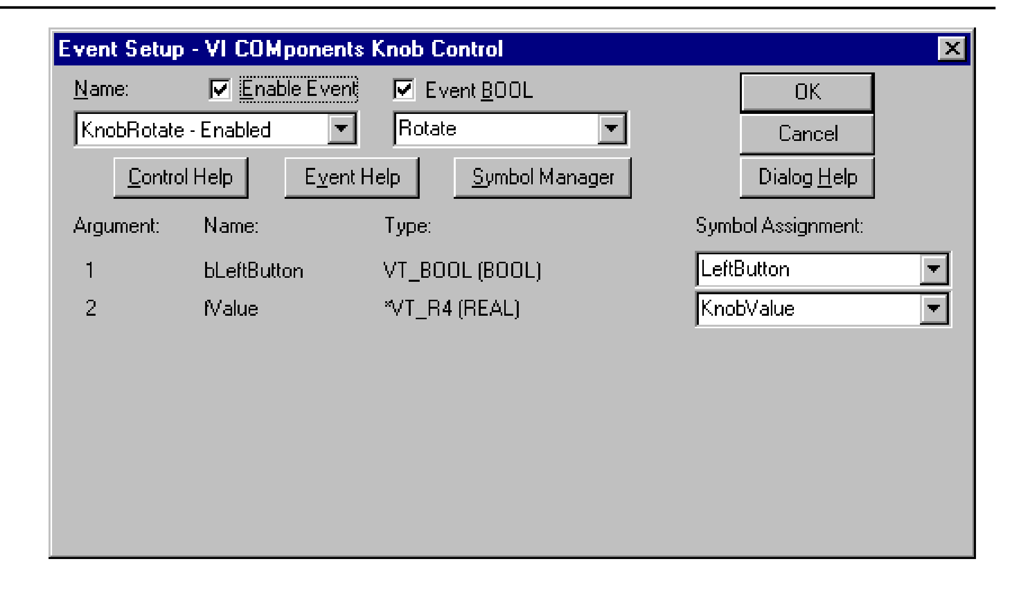
Die meisten ActiveX-Bedienelemente besitzen eine Liste mit Ereignissen, die sie dem Bedienerschnittstellencontainer melden (zum Beispiel Mausklick oder Tastenbetätigungen). Ereignisse können wahlweise mit Parametern verknüpft werden. Während ein Ereignis *im Fokus* zum Beispiel keine Parameter hat, kann ein Mausklick-Ereignis zwei ganzzahlige Parameter für die Koordinatenwerte x und y besitzen. Beim Auftreten eines Ereignisses kann die Bedienerschnittstelle zwei voneinander getrennte Dinge tun:

- Ein Boolesches Symbol auf TRUE setzen – dies ist wahlweise, wenn das Ereignis mindestens einen Parameter (Argument) besitzt. Besitzt das Ereignis keine Parameter, ist es zwingend vorgeschrieben. (Der Boolesche Wert wird durch die Bedienerschnittstelle nach dem Ereignis **nicht** auf FALSE gesetzt).
- Den Wert eines Ereignisarguments einem Symbol zuweisen – wurde einem Ereignisargument in der Konfiguration ein Symbol zugewiesen, dann wird der Wert des Arguments dem Symbol im Betrieb zugewiesen.

Ereignisse bearbeiten

1. Führen Sie einen der folgenden Schritte durch:
   - Selektieren Sie das ActiveX-Bedienelement und wählen dann *ActiveX-Ereignisse bearbeiten* aus dem Menü *Bearbeiten* oder dem Kontextmenü, oder benutzen Sie die Symbolleiste.
   - Doppelklicken Sie auf das ActiveX-Bedienelement und wählen Sie dann *Ereignisse bearbeiten* aus der nun erscheinenden Dialogbox *ActiveX-Bearbeitungsmodus auswählen*.
2. Es erscheint eine Dialogbox *Ereigniseinstellung*. Nehmen Sie die nachstehende Tabelle zur Hilfe und konfigurieren Sie die Ereignisse entsprechend den Anforderungen. Am Ende der Arbeiten klicken Sie auf *OK*, um die Konfiguration zu speichern und zum Bedienerschnittstellenmenü-Editor zurückzukehren.

Bei Bedienelementen, die keine Dialogbox *Ereigniseinstellung* besitzen, erscheint stattdessen eine entsprechende Meldung.



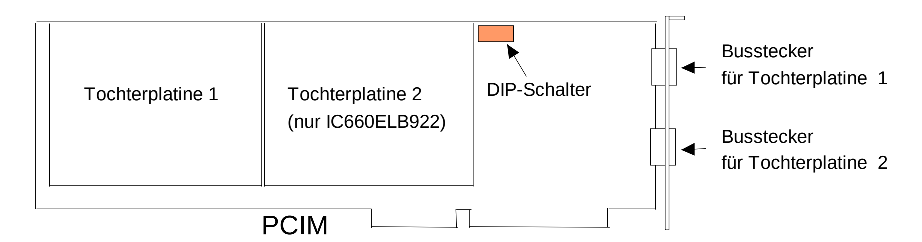
Feld	Beschreibung
Name	Wählen Sie das zu konfigurierende Ereignis aus dem Namen-Listenfeld.
Ereignis freigeben	Aktivieren Sie dieses Kontrollkästchen, um das Ereignis freizugeben. Damit das Ereignis an die Bedienerschnittstelle gemeldet werden kann, muß es hier freigegeben werden.
Ereignis BOOL	Ist dieses Kontrollkästchen aktiviert und wurde ein Boolesches Symbol zugewiesen, dann setzt das Ereignis das Boolesche Symbol im Betrieb auf TRUE. Falls erforderlich, klicken Sie auf *Symbolmanager*, um ein neues Symbol zu erzeugen.
Argumente	Akzeptiert das Ereignis Argumente, dann werden diese mit Nummer, Namen und Datentyp aufgelistet. Ereignisse sind auf 16 Argumente beschränkt. Eine Fehlermeldung erscheint, wenn das Ereignis mehr als 16 Argumente umfaßt. Für jedes Argument, dem ein Wert zugewiesen werden muß, weisen Sie ein Symbol aus seiner Symbolzuweisungsliste zu. Falls erforderlich, klicken Sie auf *Symbolmanager* , um neue Symbole zu erzeugen.
Symbolmanager	Zeigt den Symbolmanager an.
Dialoghilfe	Zeigt Hilfe zur Methodeneinstellungs-Dialogbox an.
Bedienelement-Hilfe	Zeigt Hilfe zum ActiveX-Bedienelement an (vom Bedienelemente-Hersteller geliefert).
Methoden-Hilfe	Zeigt Hilfe zur Methode des ActiveX-Bedienelements an (vom Bedienelemente-Hersteller geliefert).

Anhang
A
Das Personalcomputer-Schnittstellenmodul für Genius E/A

Dieser Anhang beschreibt das Genius® E/A Personalcomputer-Schnittstellenmodul (PCIM) von GE Fanuc. Er enthält auch Prozeduren zur Installation und Konfiguration des PCIM.

Beschreibung

Das PC-Schnittstelenmodul (PCIM), das einen Steckplatz belegt, bildet den Eintrittspunkt in das Genius E/A-System für den IBM PC. Das PCIM ist ein Modul im "AT" Stil, das für die Integration in ein vom Anwender entwickeltes Mikroprozessorsystem ausgelegt wurde. Es ist voll kompatibel zu allen Genius-Protokollen, mechanischen und elektrischen Eigenschaften und zum Kommunikations-Zeitverhalten.



Das PCIM ist lieferbar mit einer oder zwei Tochterplatine(n) (IC660ELB921/922). Jede PCIM Tochterplatine bietet eine kostengünstige 'Anzapfung' des Genius E/A-Busses. Hierdurch kann ein Hostsystem dezentrale E/A steuern und dabei die umfangreichen Diagnosefunktionen sowie die hohe Zuverlässigkeit und Störfestigkeit des Genius E/A-Systems ausnutzen. Jede Tochterplatine kann über die mit dem PCIM gelieferte Konfigurationssoftware für sich und unabhängig von den anderen konfiguriert werden.

Das PCIM wird über Kantensteckverbinder mit dem Genius Bus verbunden. Besitzt ein PCIM zwei Tochterplatinen dann können diese am gleichen Bus oder an zwei voneinander unabhängigen Bussen angeschlossen werden.

Tochterplatine

Eine PCIM-Tochterplatine ist ein Allzweck-E/A-Controller für das Genius E/A-System. Er bietet eine bequeme Möglichkeit, Geräte auf dem seriellen Genius Bus zu steuern. Die PCIM Tochterplatine führt die Organisationsaufgaben der Initialisierung und Fehlerverwaltung für bis zu 30 Busteilnehmer durch, unterhält neueste Abbilder der einzelnen von der E/A gesteuerten Geräte (unabhängig davon, ob es sich um einen Genius E/A-Block oder einen anderen Busteilnehmer handelt), und kommuniziert mit anderen Controllern auf dem Genius Bus, indem sie Hintergrundmitteilungen weitergibt, die nicht mit E/A-Befehlen oder Globaldaten zusammenhängen. Die Schnittstelle zu diesem RAM ist für den IBM Personalcomputerbus optimiert.

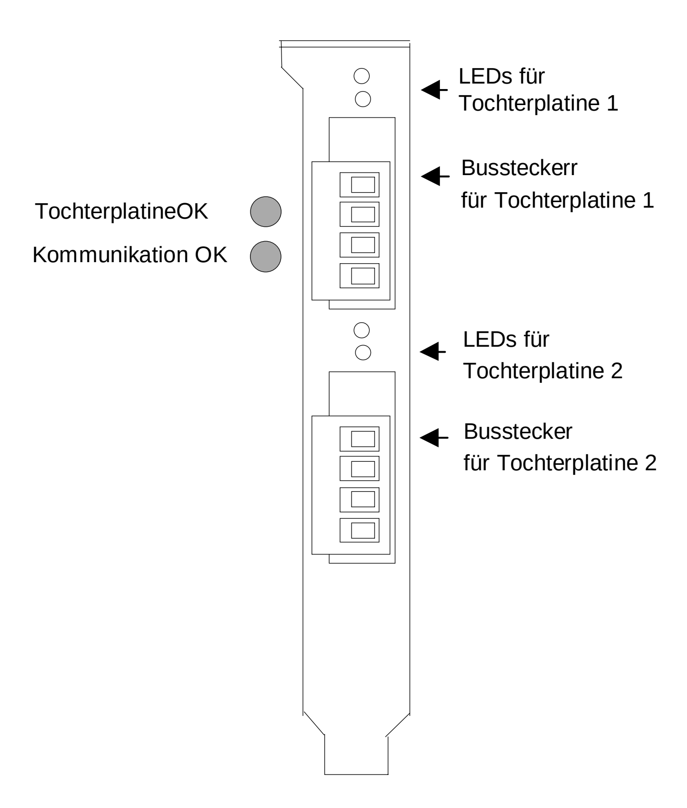
Mutterplatine

Die PCIM Mutterplatine bietet eine bequeme Möglichkeit, eine Tochterplatine mit offener Architektur, wie die PCIM Tochterplatine, an ein IBM-kompatibles Hostsystem anzuschließen. Alle Signale, die zur Kommunikation mit einer Tochterplatine notwendig sind, werden über die Mutterplatine zum Hostbus gepuffert. Zusätzlich zu den normalen Schnittstellenleitungen ermöglicht die Mutterplatine die folgenden Bedien- und Überwachungsfunktionen der Tochterplatine:

- Eine Standard 'Gerätelast' auf dem IBM Bus.
- Funktioniert in ISA-kompatiblen Rückwandplatinen.
- Erkennung eines Versorgungsspannungsabfalls.
- Einschalt-RESET-Signalfolge.
- Hostsystem-Adressendecodierung über die vollständigen PC AT Speicherabbilder.
- Ein integrierter Überwachungszeitgeber, der den Systembetrieb überwachen kann und die Tochterplatine abschaltet, wenn im Hostsystem Fehler auftreten. Hierdurch werden Konflikte auf dem Genius Bus vermieden.

Frontplatte

Für jede Tochterplatine gibt es in der PCIM-Frontplatte zwei LEDs (Modul OK und Kommunikation OK). Die nachstehende Abbildung zeigt die LEDs für die Tochterplatinen.



Die seriellen Bus-Steckverbinder für die PCIM-Tochterplatinen sitzen in Öffnungen in der Frontplatte.

Installation und Konfiguration

Zum Anschluß des PCIM an den seriellen Genius Bus müssen Sie zunächst folgende Schritte durchführen:

- Bestimmen Sie den E/A-Port und die Adressen des gemeinsam genutzten Speichers sowie die IRQ, die Sie für das PCIM benutzen.
- Stellen Sie die EEPROM E/A-Adresse an den DIP-Schaltern ein.
- Bauen Sie das PCIM in Ihren Computer ein.

- Konfigurieren Sie das PCIM mit dem PCIM Konfigurations-Dienstprogramm (PCU)
- Schließen Sie das PCIM an den seriellen Bus an.

Erforderliche Hardware

Die Hardwareanforderungen für Installation und Betrieb des PCIM sind die gleichen wie für den Betrieb der PC Control Software.

Geeignete Computer

Das PCIM wurde erfolgreich mit zahlreichen Typen von IBM PC-kompatiblen Computern getestet. Es ist 100% kompatibel zur ISA-Rückwandplatine und ermöglicht die Decodierung von Hostsystemadressen über die vollständigen PC AT Speicherabbilder. Es war jedoch nicht möglich, das PCIM mit allen im Handel erhältlichen Computern zu testen. Es kann daher nicht zugesichert werden, daß das PCIM in jedem beliebigen Hostcomputer zufriedenstellend arbeitet.

Computer-Betriebsmittel, die Sie reservieren müssen

Zur Installation und Konfiguration des PCIM müssen Sie bestimmte Computersystem-Betriebsmittel (E/A-Ports, Speicher und IRQ) reservieren. Dies erfolgt über DIP-Schalter (EEPROM-Adresse) und das PCIM Konfigurations-Dienstprogramm. Ehe Sie mit der eigentlichen Installation und Konfiguration beginnen, sollten Sie festlegen, welche Adressen Sie benutzen können, ohne dadurch mit den bereits in Ihrem Computer installierten Geräten in Konflikt zu kommen.

Notieren Sie diese Informationen, um sie bei der Konfiguration des PCIM nachschlagen zu können.

Sie müssen folgende Betriebsmittel für das PCIM reservieren:

- Ein E/A-Port für die mit den DIP-Schaltern eingestellte EEPROM-Adresse (gleicher Wert wie die über das PCIM Konfigurations-Dienstprogramm eingestellte Controller-Portadresse).
- Ein Block mit vier E/A-Ports für *jede* GENI Tochterplatine. Dieser Block muß im Bereich 100 bis 3E4 (hexa) liegen und auf einer 4-Byte-Grenze (0, 4, 8, C hexadezimal) beginnen.
- Ein Block von 4000 (hexa) gemeinsam genutzten Speichers für *jede* GENI Tochterplatine. Dieser Block muß auf einer 16-kB-Grenze (0000, 4000, 8000, C000 hexa) liegen.
- Eine freie IRQ (3, 4, 5, 6, 9, 10, oder 11), die von allen GENI Tochterplatinen in Ihrem System gemeinsam genutzt wird.

Vorgeschlagene Adressen zur Vermeidung von Konflikten

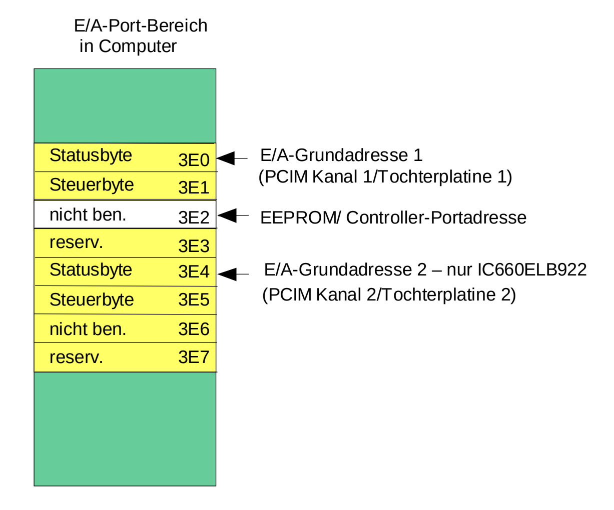
Das PCIM Konfigurations-Dienstprogramm (PCU) trägt automatisch Vorschlagswerte für Controllerport, E/A-Port und Speichergrundadressen sowie den IRQ-Wert ein. Diese Werte werden aus den nachstehend gezeigten Voschlagswerten entnommen. Um Konflikte zu vermeiden müssen Sie auf jeden Fall die Systemkonfiguration Ihres Computers überprüfen.

EEPROM-Adresse (gleicher Wert wie Controller-Portadresse)

Adresse eines 1-Byte E/A-Ports, die über die PCIM DIP-Schalter (EEPROM-Adresse) eingestellt und in der PCU (Controller-Portadresse) konfiguriert wird. Dieser E/A-Port wird von der PCU dazu benutzt, PCIM-Konfigurationsdaten in den EEPROM auf dem PCIM zu schreiben und den Anwendungen bei Bedarf den Zugriff auf die Daten zu ermöglichen.

Diese E/A-Portadresse ist unabhängig von den E/A-Port, die von den Tochterplatinen benötigt werden (siehe E/A-Grundadresse). Wir schlagen jedoch vor, die Adresse in einem ungenutzten Teil des für die erste GENI Tochterplatine auf dem PCIM reservierten E/A-Portbereichs einzustellen (d.h. um 2 höher als die E/A-Grundadresse für die Tochterplatine). Dieses Byte wird vom PCIM nicht benutzt und minimiert die Wahrscheinlichkeit von Adreßkonflikten. ***Siehe nachstehende Abbildung.***

Beispiel: Wenn Sie 3E0 als E/A-Grundadresse für die erste GENI Tochterplatine (Kanal 1) des PCIM auswählen, dann benutzen Sie 3E2 für die EEPROM-Adresse (Controllerport).



E/A-Grundadresse

Anfangsadresse des 4-Byte E/A-Portbereichs, den Sie in Ihrem Computer für die einzelnen GENI Tochterplatine (Kanäle) im PCIM reservieren müssen. Diese Adresse muß auf einer 4-Byte-Grenze liegen.

Die nachstehenden Adressen können normalerweise problemlos verwendet werden:

3E0	348	2E0	228
3E4	34C	2E4	22C
340		220	
344		224	

Überprüfen Sie die Systemkonfiguration Ihres Computers auf mögliche Konflikte.

Speichergrundadresse

Anfangsadresse des 16-kB-Speicherblocks, den Sie in Ihrem Computer für die einzelnen GENI Tochterplatine (Kanäle) im PCIM reservieren müssen. Diese Adresse muß auf einer 16-kB-Grenze liegen.

Die nachstehenden Adressen können normalerweise problemlos verwendet werden, wenn Ihr Computer nicht die angegebene Hardware oder Software verwendet:

B0000	Verwendet für monochromen Videospeicher
B4000	Verwendet für monochromen Videospeicher
C8000	Normalerweise OK
CC000	Verwendet für die parallele WSI-Karte von GE Fanuc
D0000	Verwendet für EMS-Speicher
D4000	Verwendet für EMS-Speicher
D8000	Normalerweise OK
DC000	Normalerweise OK
E0000	Normalerweise OK
E4000	Normalerweise OK
E8000	Normalerweise OK
EC000	Normalerweise OK

Überprüfen Sie die Systemkonfiguration Ihres Computers auf mögliche Konflikte.

IRQ

Interrupt für das PCIM.

Überprüfen Sie, ob IRQ 10 oder 11 verwendet wird. Sie können die folgenden IRQs für das PCIM konfigurieren und gemeinsam für alle GENU Tochterplatinen in Ihrem System verwenden: 3, 4, 5, 6, 9, 10, oder 11. *Überprüfen Sie die Systemkonfiguration Ihres Computers auf mögliche Konflikte.*

Nachstehend sehen Sie einige gebräuchliche IRQ-Verknüpfungen:

IRQ3 Serieller Port COM2:
IRQ4 Serieller Port COM1:
IRQ5 Netzwerk/LPT2:
IRQ6 Floppy Disk

Betriebsmittelkonflikte bei E/A-Port, Speicher und Interrupt feststellen

Bei der Installation und Konfiguration des PCIM weisen Sie Werte zu für EEPROM-Adresse, Controller-Portadresse, Speichergrundadresse, E/A-Grundadresse und IRQ. Sie müssen für diese Parameter Werte auswählen, die nicht mit denen kollidieren, die für andere in Ihrem Computer eingebaute Geräte verwendet werden. Dieses sind die von den Geräten in Ihrem Computer verwendeten Betriebsmittel.

Windows NT 4.0 Betriebsmittel

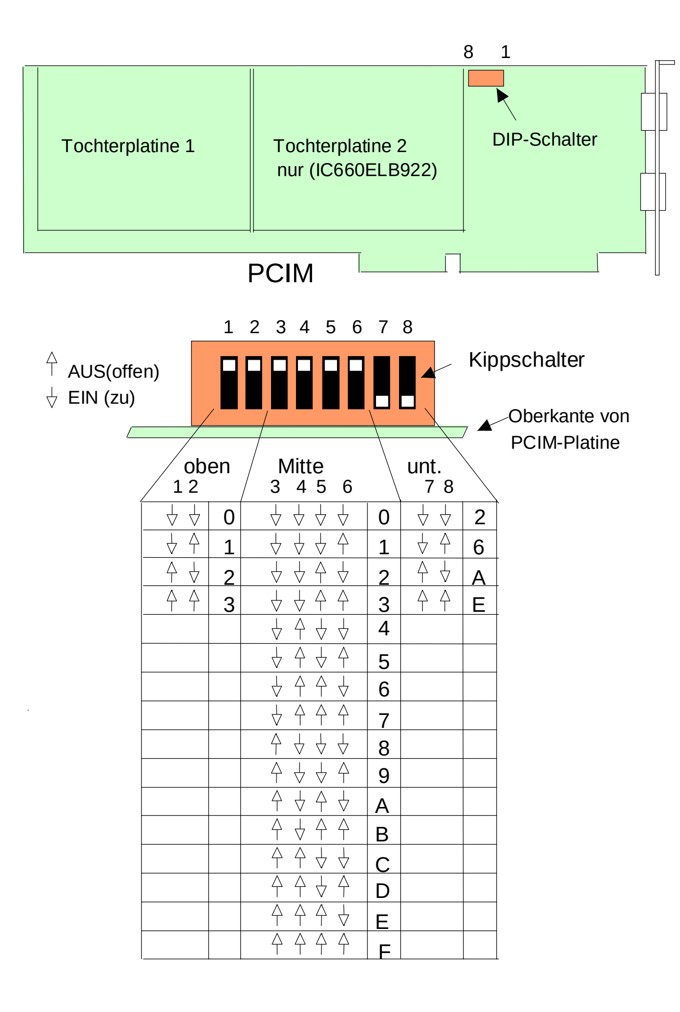
Melden Sie sich an unter einem Konto mit Systemadministrator-Privilegien.

1. Drücken Sie die Taste Start.
2. Wählen Sie im Startmenü die Optionen Programme, Administrative Werkzeuge, Windows NT-Diagnosefunktionen.
3. Klicken Sie auf die Karteikarte Betriebsmittel. Schauen Sie dann in den Dialogboxen E/A-Port, Speicher und IRQ nach unbenutzten Adressen.
4. Suchen Sie für jede GENI Tochterplatine einen Block mit 4 unbenutzten (nicht aufgelisteten) E/A-Portadressen. Der erste Port in dem Block ist die *E/A-Grundadresse* für einen im PCIM Konfigurations-Dienstprogramm konfigurierten Kanal.
5. Suchen Sie für jede GENI Tochterplatine einen Block mit 4000 unbenutzten (nicht aufgelisteten) Speicheradressen. Die Anfangsadresse dieses Blocks ist die *Speichergrundadresse* für einen im PCIM Konfigurations-Dienstprogramm konfigurierten Kanal.
6. Suchen Sie eine freie (nicht aufgelistete) IRQ für den im PCIM Konfigurations-Dienstprogramm konfigurierten *Interrupt*.

DIP-Schalter auf dem PCIM

Der über die DIP-Schalter eingestellte E/A-Port wird vom PCU dazu benutzt, PCIM-Konfigurationsdaten in das EEPROM auf dem PCIM zu schreiben und Anwendungen bei Bedarf den Zugriff auf diese Daten zu ermöglichen.

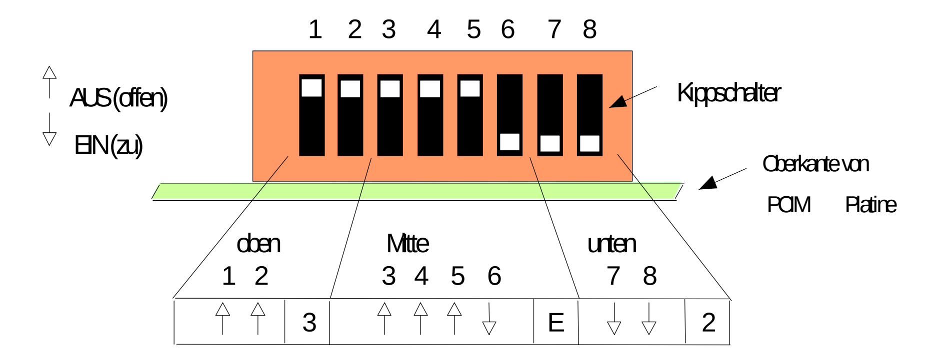
Die Schalterpositionen sind von 1 bis 8 durchnumeriert. Stellen Sie mit den Schaltern 1 und 2 die obere Hexadezimalstelle ein. Mit den Schaltern 3, 4, 5 und 6 stellen Sie die mittlere Hexadezimalstelle ein und mit den Schaltern 7 und 8 die untere Hexadezimalstelle.



oben 1 2		Mitte 3 4 5 6		unt. 7 8	
↓ ↓	0	↓ ↓ ↓ ↓	0	↓ ↓	2
↓ ↑	1	↓ ↓ ↓ ↑	1	↓ ↑	6
↑ ↓	2	↓ ↓ ↑ ↓	2	↑ ↓	A
↑ ↑	3	↓ ↓ ↑ ↑	3	↑ ↑	E
		↓ ↑ ↓ ↓	4		
		↓ ↑ ↓ ↑	5		
		↓ ↑ ↑ ↓	6		
		↓ ↑ ↑ ↑	7		
		↑ ↓ ↓ ↓	8		
		↑ ↓ ↓ ↑	9		
		↑ ↓ ↑ ↓	A		
		↑ ↓ ↑ ↑	B		
		↑ ↑ ↓ ↓	C		
		↑ ↑ ↓ ↑	D		
		↑ ↑ ↑ ↓	E		
		↑ ↑ ↑ ↑	F		

Schaltereinstellung - Beispiel

Wollen Sie die erste GENI Tochterplatine (Kanal 1) auf 3E0 (hexa) setzen, dann sollten Sie den EEPROM-DIP-Schalter auf 3E2 einstellen. In diesem Beispiel wird gezeigt, wie Sie die EEPROM-Adresse auf 3E2 einstellen können.



PCIM im Computer einbauen

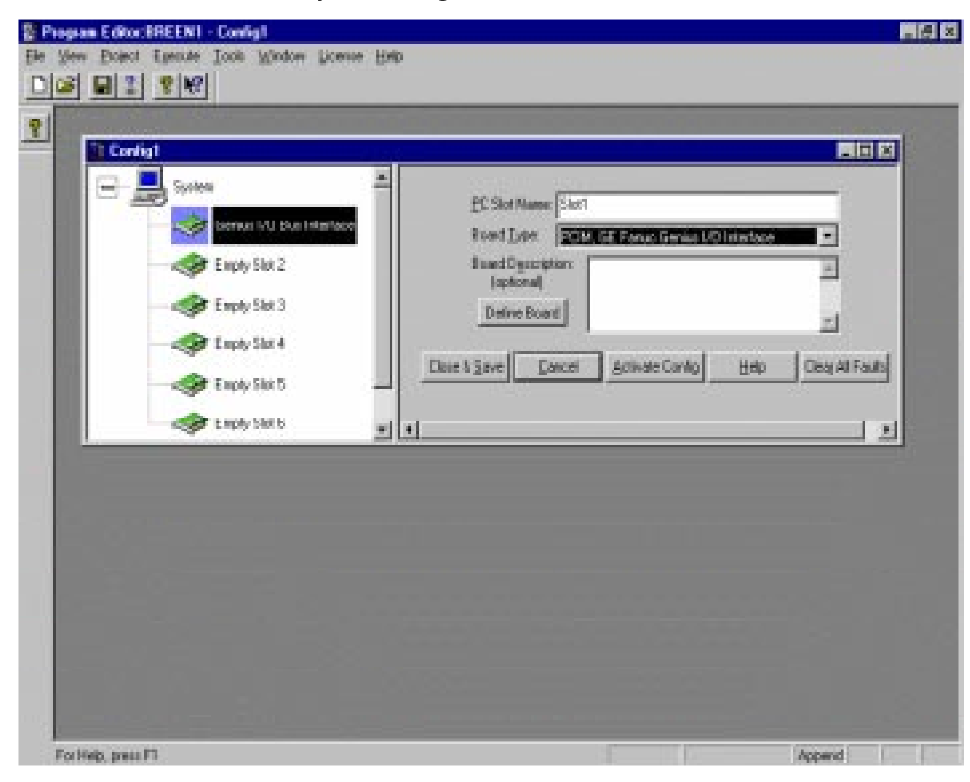
1. Schalten Sie den Hostcomputer aus und trennen Sie ihn von der Spannungsquelle.
2. Bauen Sie das PCM entsprechend den Herstelleranweisungen für Zusatzkarten ein.
3. Schließen Sie den Bus an das PCIM an (diesen Schritt können Sie auch nach der Konfiguration des PCIM durchführen).
4. NIEMALS
   - das PCIM an einer Stelle einbauen, an der die freie Luftströmung behindert wird.
   - das PCIM mit einem Abstand von weniger als 3,18 mm (1/8 Zoll) zu anderen Platinen oder Chassiskomponenten einbauen.
   - Klebstoffe oder konforme Beschichtungen auf PCIM-Teilen verwenden.

PCIM konfigurieren

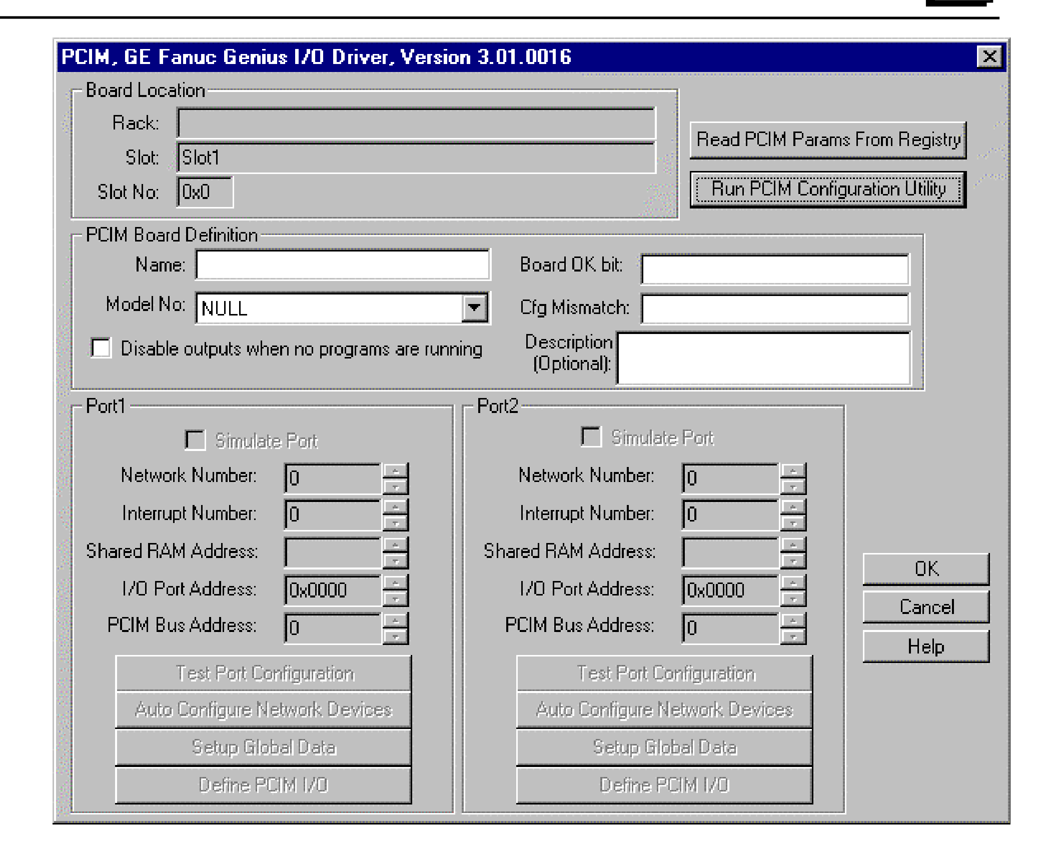
Ehe Sie das PCIM konfigurieren müssen Sie die verfügbaren Adressen für E/A-Port und gemeinsam benutzen Speicher sowie die verwendete IRQ festlegen. Siehe hierzu den Abschnitt "Computer-Betriebsmittel, die reserviert werden müssen" weiter vorne.

PCIM konfigurieren:

1. Starten Sie PC Control, Programmeditor.
2. Wählen Sie im Menü "Datei" die Option "Neu" oder "Konfiguration öffnen", um eine neue Systemkonfiguration zu erstellen oder eine bestehende zu öffnen. Hierauf erscheint das Systemkonfigurationsmenü.



3. Klicken Sie unter dem Symbol "System" auf ein Symbol "Leerer Steckplatz".
4. Wählen Sie im Feld "Modul" PCIM aus der Dropdown-Liste und klicken dann auf die Schaltfläche "Moduldefinition". Hierauf erscheint die Dialogbox "PCIM-Genius E/A-Treiber".



5. Starten Sie das PCIM Konfigurations-Dienstprogramm. Klicken Sie hierzu auf die Schaltfläche "PCIM Konfigurations-Dienstprogramm aktivieren". Im PCU konfigurieren Sie eine Reihe von Feldern. Verwenden Sie hierbei in der von Ihnen vorbereiteten Liste verfügbarer Adressen und IRQs. Die Online-Hilfe hilft Ihnen bei Bedarf während der Konfiguration.

6. Nachdem Sie das PCIM konfiguriert und das PCU verlassen haben, klicken Sie auf die Schaltfläche "PCIM Parameter aus Registrierdatenbank lesen". Hierauf erscheint ein Fenster mit einer Liste der konfigurierten PCIMs.

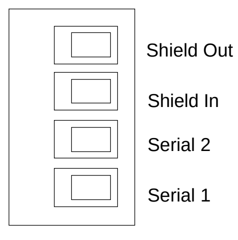
7. Wählen Sie das entsprechende PCIM aus und klicken dann auf OK. Hierdurch werden die Portparameter in der Dialogbox "Genius E/A-Treiber" mit den konfigurierten Werten von Interruptnummer (IRQ), gemeinsam genutzter RAM-Adresse (Speichergrundadresse), E/A-Portadresse, (E/A-Grundadresse), und PCIM Busadresse (serielle Busadresse) aktualisiert.

8. Klicken Sie abschließend auf die Schaltfläche "Portkonfiguration testen". Nun müßten die LEDs "Modul OK" und "Kommunikation OK" aufleuchten.

Bei Problemen können Sie im Abschnitt "Fehlersuche" nachschlagen.

PCIM am Bus anschließen

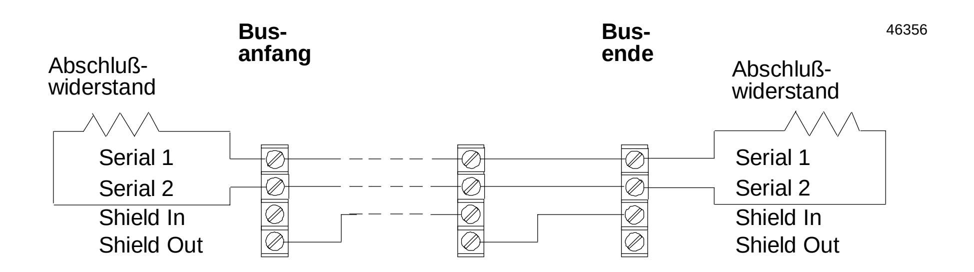
Die Teilnehmer können in beliebiger physikalischer Reihenfolge am Bus plaziert werden. Jeder Steckverbinder am PCIM besitzt vier Klemmen für das Buskabel (Serial 1, Serial 2, Shield In und Shield Out). Beachten Sie, daß die Anordnung dieser Klemmen am PCIM-Steckverbinder anders ist wie bei anderen Busteilnehmern (z.B. E/A-Blöcke).



Diese Klemmen können zwei Drähte mit einem Querschnitt von 0,54 mm^2 plus einen Anschlußdraht eines 0,25 Watt Widerstands (wahlweise für Busabschluß) aufnehmen. Der kleinste empfohlene Drahtquerschnitt ist 0,36 mm^2 .

Verbinden Sie bei jedem Steckverbinder die Klemme "Serial 1" mit den Klemmen "Serial 1" des vorhergehenden und des nachfolgenden Gerätes. Verbinden Sie bei jedem Steckverbinder die Klemme "Serial 2" mit den Klemmen "Serial 2" des vorhergehenden und des nachfolgenden Gerätes. Besitzt ein PCIM zwei Tochterplatinen dann können diese am gleichen Bus oder an zwei unterschiedliche Busse angeschlossen werden.

Bei jedem Steckverbinder muß der Anschluß "Shield In" mit "Shield Out" des vorhergehenden Gerätes verbunden werden. Beim ersten Gerät am Bus kann "Shield In" offen bleiben. Beim letzten Gerät am Bus kann "Shield Out" offen bleiben.



Beim Anschluß der Busleitung sollten Sie nicht mehr als 5 cm des Drahtes abisolieren. Für zusätzlichen Schutz sollten Sie die einzelnen Ableitungsadern der Abschirmung mit Isolierschläuchen versehen. Hierdurch verhindern Sie einen Kontakt zwischen "Shield In" und "Shield Out".

Busabschluß

Ein Bus muß an beiden Enden durch eine für den entsprechenden Kabeltyp geeigneten Impedanz abgeschlossen werden. Die Impedanz beträgt 75, 100, 120 oder 150 Ohm. Liegt an einem Ende des Busses ein PCIM-Steckverbinder, dann müssen Sie den passenden Abschlußwiderstand an den Klemmen "Serial 1" und "Serial 2" anschließen. In *Genius E/A-System und Datenübertragung, Anwenderhandbuch* sind die passenden Abschlußwiderstände für die einzelnen empfohlenen Buskabeltypen aufgelistet.

PCIM vom Bus wegnehmen

Die Bus-Steckverbinder des PCIM sind abnehmbar. Sie können bei laufendem System abgenommen werden, ohne daß dadurch die Datensicherheit auf dem Bus gefährdet wird. Zum Abnehmen halten Sie den Bus-Steckverbinder vorsichtig oben und unten und ziehen ihn vom PCIM weg. Ist ein im Betrieb befindliches Kabel am Bus angeschlossen, müssen Sie sorgfältig darauf achten, daß die Busadern sich nicht gegenseitig oder andere leitenden Gegenstände berühren. Legen Sie den Steckverbinder niemals auf eine leitende Oberfläche.

Solange der Bus in Betrieb ist dürfen *einzelne Busadern* nie aus den Steckverbinderklemmen entfernt werden. Die sich hieraus ergebenden unzuverlässigen Daten auf dem Bus könnten gefährliche Steuerungssituationen hervorrufen.

A

Technische Daten

Bestellnummern	
Einkanal-PCIM	IC660ELB921
Zweikanal-PCIM	IC660ELB922
LEDs (2 für jede Tochterplatine)	GENI OK, COMMS OK (Kommunikation OK)
Elektrische Daten	
Leistungsbedarf	5 V DC +/– 10%, 400 mA (Maximum)
Busbelastung	1 LS TTL-Last pro Eingangsleitung
Bus-Ansteuerleistung	10 LS TTL-Lasten pro Ausgangsleitung
Mechanische Daten	
PCIM-Modultyp	Module im "AT"-Stil, 1 Steckplatz breit
Anschluß für Handmonitor	Externer Steckverbinder mit HHM- und Busklemmen
Serieller Busanschluß	Modulkantenklemmen oder externer Steckverbinder. An Modulkantenklemmen können zwei Drähte mit einem Querschnitt von 0,55 mm^2 oder drei Drähte mit einem Querschnitt von 0,36 mm^2 angeschlossen werden.
Host-Rückwandplatinenschnittstelle	100% ISA-kompatibel
Speicheranforderungen	
Mutterplatine	4 Bytes
Pro Tochterplatine	16 kB
Umgebungsbedingungen – Betrieb	
Temperatur	0 bis 60 Grad C (Umgebungstemperatur am Modul)
Luftfeuchtigkeit	5% bis 95% nicht kondensierend
Höhe	10.000 Fuß
Vibration	5,1 mm Auslenkung bei 5 bis 10 Hz 1 G 10 bis 200 Hz
Stoß	5 G, 10 ms Dauer nach MIL–STD 810C, Methode 516.2
Umgebungsbedingungen – Lagerung –Transport	
Temperatur	–40 bis 125 Grad C (Umgebungstemperatur am Modul)
Luftfeuchtigkeit	5% bis 95% nicht kondensierend
Höhe	40,000 Fuß
Vibration	5,1 mm Auslenkung bei 5 bis 10 Hz 1 G 10 bis 200 Hz
Stoß	5 G, 10 ms Dauer nach MIL–STD 810C, Methode 516.2

Elektrische Daten des PCIM

Stromversorgung

Zur Logikversorgung benötigt das PCIM eine 5 V Gleichspannungsquelle. Das PCIM funktioniert nur dann korrekt, wenn die Versorgungsspannung nicht mehr als 10% über oder unter dem Nennwert liegt (nicht weniger als 4,5 V und nicht mehr als 5,5 V). Das PCIM mit einer Tochterplatine (Einkanal-PCIM) zieht typisch 1,0 A. Das PCIM mit zwei Tochterplatinen (Zweikanal-PCIM) zieht typisch 1,5 A.

Buslasten/Ansteuerleistung

Alle Eingangsleitungen zum PCIM stellen nicht mehr als eine Standard-LSTTL-Last zum Host-Schnittstellensteckverbinder dar.

Alle Ausgangsleitungen vom PCIM können 10 Standard-LSTTL-Lasten ansteuern. Mit Ausnahme der Leitungen /INT und /PCIM OK sind alle Leitungen Tristate-Ausgänge. Die /INT Leitung ist ein Open-Collector-Ausgang, der mit einem einzelnen Interrupteingang hardwaremäßig geODERt werden kann. /PCIM OK und /COMM OK sind Open-Collector-Typen mit TRUE auf L-Pegel und eingebauter 10-mA-Strombegrenzung, mit denen LEDs direkt betrieben werden können.

Alle Eingangssignale vom Hostsystem zum PCIM sehen für den Host wie eine LSTTL-Last aus. Diese Signale sind TTL-kompatibel und schalten bei TTL-Pegeln.

Die Steuerungs-Ausgangssignale zum Host sind Open–Collector LSTTL-Treiber mit 10 k Pull-Up-Widerständen, die bei Aufrechterhaltung einer Ausgangsspannung von 0,4 V oder weniger einen Strom von 4 mA aufnehmen können.

Der Datentransceiver ist ein Tristate-LSTTL-Gerät, das 12 mA liefern oder aufnehmen kann bei VOL = 0,4 V und VOH = 2,0 V.

Das PCIM ist 100% kompatibel zu ISA-Rückwandplatinen.

Signalaufbereitung

Das PCIM besitzt zwei Steckverbinder, die zugänglich sind, wenn das PCIM in einem PC-Chassis eingebaut ist. Beide Steckverbinder sind für Standardanschlüsse mit verdrillten Paaren an einen seriellen Bus geeignet.

Der Handmonitor kann über ein Schnittstellenkabel an die getrennten Genius Steckverbinder angeschlossen werden.

Alle über die beiden Steckverbinder eingehenden Leitungen sind entweder potentialgetrennt oder induktiv begrenzt, um das PCIM vor Spannungsspitzen oder irrtümlich angelegten hohen Spannungen an den seriellen Busanschlüssen zu schützen.

Fehlersuche

Wie beim Austesten eines Programms erfolgt die Fehlersuche bei Hardware/Firmware über logisches Durchdenken der Funktionen der einzelnen Systemteile und des Zusammenspiels dieser Funktionen. Ein grundsätzliches Verständnis der verschiedenen Anzeigeleuchten hilft dabei, das Problem schnell auf das PCIM, eine Buscontroller, ein E/A-Chassis, einen E/A-Block oder die CPU einzugrenzen.

Tritt ein Problem auf, muß das gesamte System betrachtet werden. CPU, Hostcomputer, E/A-Blöcke und externe Teilnehmer, die am Genius E/A-System angeschlossen sind oder von diesem gesteuert werden müssen alle betriebsfähig und richtig angeschlossen sein. Überprüfen Sie sorgfältig alle Kabelanschlüsse sowie alle Schraub- und Lötanschlüsse.

Modultausch

Tritt ein Problem auf, kreisen Sie es zunächst bis zu einer größeren Baueinheit ein. Anschließend finden Sie innerhalb dieser Einheit das defekte Modul heraus. Das defekte Modul wird dann aus einem vor Ort als Ersatzteile vorgehaltenen Satz doppelter Module heraus ausgewechselt. Auf diese Weise ist Ihre Fertigung oder Ihr System wieder schnell betriebsbereit.

Das defekte Modul kann über die normalen Kanäle zur Reparatur unter oder außerhalb der Gewährleistung zurückgeschickt werden, ohne daß dabei Ihre Fertigung oder Ihr System für einen längeren Zeitraum stillsteht. Das Austauschkonzept minimiert die Ausfallzeiten auf Minuten (im Gegensatz zu Tagen). Die möglichen Einsparungen übertreffen bei weitem die vergleichsweise geringen Kosten für Ersatzmodule.

Falls Sie mit Ihrem ursprünglichen System keine Ersatzmodule erworben haben, sollten Sie mit Ihrem autorisierten GE Fanuc Distributor Kontakt aufnehmen und dies jetzt nachholen. Mit Hilfe dieses Handbuches und dem Personal Ihres örtlichen autorisierten GE Fanuc Distributors können Sie praktisch jedes auftretende Problem finden und beheben.

PCIM Fehlersuche

LEDs

Ein Fehler, der zu Störungen des PCIM führt, kann normalerweise über die Status-LEDs auf dem PCIM lokalisiert werden. Normalerweise leuchten die Status-LEDs. Leuchtet eine LED nicht, dann führen Sie die Fehlersuche in den in diesem Abschnitt beschriebenen Schritten durch.

Anzeige	Status	Definition
BOARD OK	EIN	Am PCIM liegt Spannung an (das Modul funktioniert nur richtig, wenn es mit der geforderten Spannung versorgt wird), und der moduleigene Selbsttest wurde fehlerfrei durchlaufen.
	AUS	Der Überwachungszeitgeber ist abgelaufen und zeigt einen Modulfehler, eine falsche Adreßzuweisung oder eine auf L-Pegel liegende /RST Eingangsleitung an.
COMM OK	EIN	Die Versorgungsspannung liegt an, die Kommunikationshardware des Controllers funktioniert und er kann in jedem Buszyklus Daten senden (erhält das Token).
	AUS	(oder BLINKEN) Es ist ein Fehler aufgetreten in der Kommunikationshardware oder beim Zugriff auf den seriellen Genius Bus.

Fehlereingrenzung und Fehlerbehebung

Sind zwar die Status-LEDs im richtigen Zustand, aber der Bus funktioniert nicht richtig, dann können die folgenden Störungszustände eventuell das Problem beschreiben. Folgen Sie in diesem Fall den bei dem entsprechenden Störungsfall beschriebenen Schritten.

- Das PCIM ist eingesteckt und eingeschaltet und eine LED leuchtet nicht.

  Board OK ist AUS und Comm OK ist EIN -

  - Prüfen Sie die mit der Konfigurationssoftware eingegebenen Parameter.

  Die BOARD OK LED leuchtet nicht.

  - Überprüfen Sie den Sitz des PCIM in der Host-Rückwandplatine und schauen Sie nach, ob alle Steckerstifte in Ordnung sind.

  Scheint alles in Ordnung zu sein, kann ein Hardwarefehler vorliegen. Wechseln Sie das PCIM aus.

  Board OK ist EIN und Comm OK ist AUS -

  - Prüfen Sie, ob Kabeltyp und Kabellänge den Anforderungen entsprechen (siehe *Genius E/A-System und Datenübertragung, Anwenderhandbuch*, GEK-90486-1).
  - Überprüfen Sie, ob an beiden Enden des Busses die richtigen Abschlußwiderstände angeschlossen sind (siehe Genius E/A-System und Datenübertragung, Anwenderhandbuch, GEK-90486-1).
  - Stellen Sie fest, ob die serielle Busverdrahtung als verkettete Struktur ausgeführt wurde.
  - Stellen Sie sicher, daß die Buskabel nicht in der Nähe von Hochspannungsleitungen liegen.
  - Suchen Sie nach defekten Kabeln.

  Beide LEDs sind AUS -

  - Überprüfen Sie Sitz und Spannungsversorgung des PCIM.

  Beide LEDs blinken gemeinsam -

  - Wahrscheinlich wurden zwei Teilnehmer am Bus mit der gleichen Teilnehmernummer (serielle Busadresse) konfiguriert.

  Mit HHM überprüfen.

- Wiederholte Busfehler
  - Überprüfen Sie, ob die Kabelabschirmung richtig angeschlossen und geerdet ist (siehe Genius E/A-System und Datenübertragung, Anwenderhandbuch, GEK-90486-1).
  - Ziehen Sie das Bus-Datenkabel vom PCIM ab, schauen in den Teilnehmernummer-Blättern nach, mit denen Sie das System konfiguriert haben, und vergleichen Sie mit dem HHM die Teilnehmernummern und E/A-Adressen.
  - Scheint alles in Ordnung zu sein, wechseln Sie das PCIM aus.
- System wird abgeschaltet mit Paritätsfehlern.

  Doppelt vergebene oder sich überlappende PCIM/E/A-Referenzen.

  - Doppelter Eingang am gleichen Bus.
  - Die Eingangsreferenzen von anderen PCIMs überlappen.
- Busfehler -PCIM kann nicht hochlaufen.
  - Serial 1/Serial 2 gekreuzt

- Sporadischer oder vollständiger Ausfall des Datenverkehrs.
  - Übertragungsgeschwindigkeiten nicht einheitlich

  Schalten Sie die Blöcke einzeln ein und überprüfen Sie die Übertragungsgeschwindigkeit. Eine Änderung der Übertragungsgeschwindigkeit eines Blocks wird erst wirksam, wenn der betreffende Block aus- und wieder eingeschaltet wird.

- Keine Globaldaten.
  - Zielteilnehmer ist offline.

  Überprüfen Sie, ob das Ziel online ist.

- Datagram erfolglos abgeschlossen.
  - Zielteilnehmer ist offline.

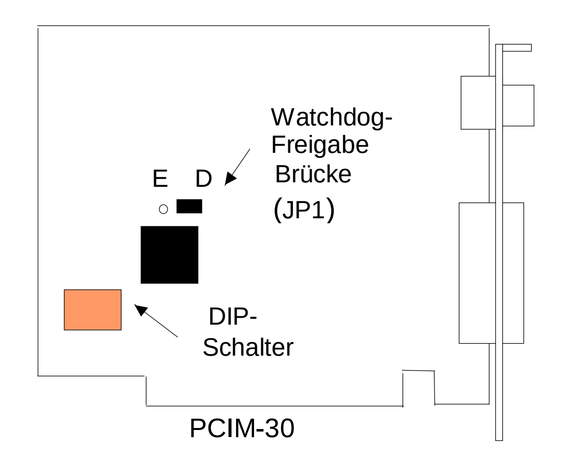
Überprüfen Sie, ob das Ziel online ist.

Anhang
B

Personalcomputer-Schnittstelle (PCIF) für Series 90-30 E/A

Die Personalcomputer-Schnittstelle für Series 90-30 E/A (IC693PIF301) ermöglicht den E/A-Zugriff auf bis zu vier Series 90-30 Erweiterungschassis oder dezentrale Chassis. Die Karte, die einen halben Steckplatz belegt, ist für IBM PC/AT-kompatible Computer ausgelegt. Die PCIF-30 unterstützt alle digitale und analoge E/A-Module sowie einige Spezialmodule.

Für die Verbindung zwischen Modul und Chassis werden die gleichen Kabel wie beim Standard-Erweiterungssystem oder dezentralen System der Series 90-30 verwendet. Das Module besitzt auch einen Hardware-Überwachungszeitgeber, der alle Ausgänge auf Rücksetzzustand fixiert und ein Relais öffnet, wenn die PC mit der Aktualisierung der Ausgänge aufhört.



Einbau des PCIF-30

Beim Einbau des PCIF-30 müssen Sie im E/A-Port-Bereich Ihres Computers acht zusammenhängende E/A-Ports reservieren. Stellen Sie hierzu die Moduladresse über die DIP-Schalter ein. Ehe Sie mit der eigentlichen Installation beginnen, sollten Sie festlegen, welche Ports Sie benutzen können, ohne dadurch mit E/A-Ports in Konflikt zu kommen, die den bereits in Ihrem Computer installierten Geräten zugewiesen wurden.

Anfangen

1. Lesen Sie zuerst den Abschnitt "Vorgeschlagene Adressen zur Vermeidung von Konflikten".
2. Stellen Sie fest, ob die vorgeschlagenen Adressen (Betriebsmittel) in Ihrem System verfügbar sind.
3. Stellen Sie die DIP-Schalter auf der PCIF-30 ein und bauen Sie das Modul in Ihren Computer ein.
4. Setzen Sie die Brücke zur Freigabe des Überwachungszeitgebers auf "D" (disabled= gesperrt).

Vorgeschlagene Adressen zur Vermeidung von Konflikten

Moduladresse

Anfangsadresse des 8-Byte E/A-Portbereichs, den Sie in Ihrem Computer für das PCIF-30 reservieren müssen. Diese Adresse muß auf einer 8-Byte-Grenze liegen.

Die nachstehenden Adressen können normalerweise problemlos verwendet werden:

310	348	2E0	228
318	34C	2E4	22C
340	3E0	220	
344	3E4	224	

Überprüfen Sie die Systemkonfiguration Ihres Computers auf mögliche Konflikte.

Betriebsmittelkonflikte bei E/A-Port feststellen

Bei der Installation des PCIF-30 weisen Sie der Moduladresse über die DIP-Schalter einen Wert zu. Sie müssen für diese Adresse einen Wert auswählen, der nicht mit denen kollidieren, die für andere in Ihrem Computer eingebaute Geräte verwendet werden. Dieses sind die von den Geräten in Ihrem Computer verwendeten Betriebsmittel.

Windows NT 4.0 Betriebsmittel

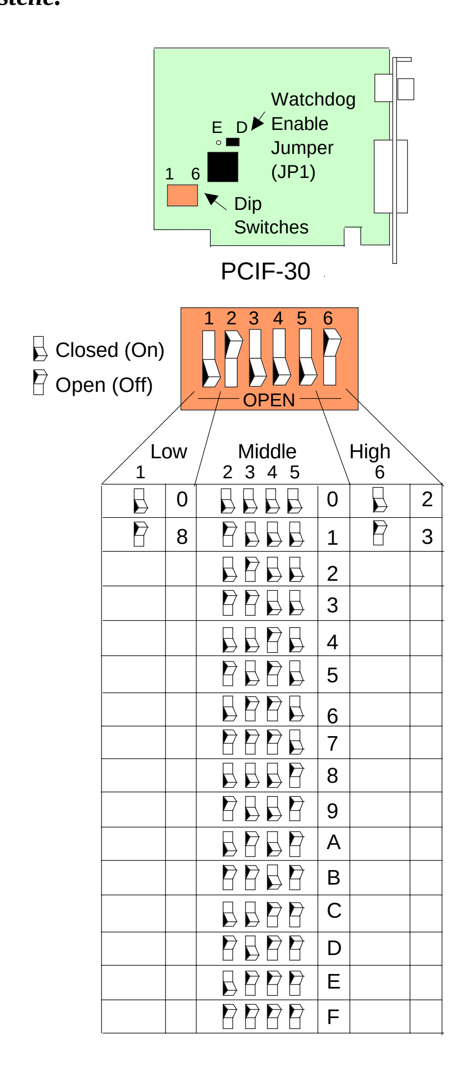
Melden Sie sich an unter einem Konto mit Systemadministrator-Privilegien.

1. Drücken Sie die Taste Start.
2. Wählen Sie im Startmenü die Optionen Programme, Administrative Werkzeuge, Windows NT-Diagnosefunktionen.
3. Klicken Sie auf die Karteikarte Betriebsmittel. Schauen Sie dann in der E/A-Port- Dialogbox nach unbenutzten Adressen.
4. Suchen Sie einen Block von 8 unbenutzten (nicht aufgelisteten) E/A-Portadressen für das PCIF-30. Der erste Port in dem Block ist die Moduladresse.

DIP-Schalter und Überwachungszeitgeber-Brücken auf PCIF-30 einstellen

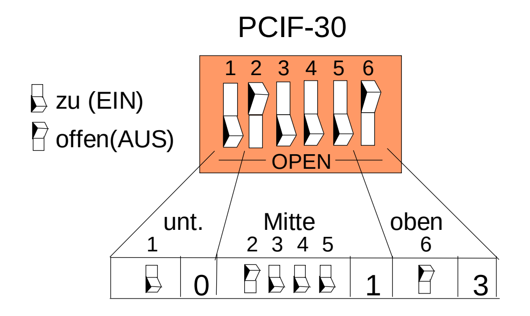
Die mit den DIP-Schaltern eingestellte Moduladresse wird zur Reservierung der für den Betrieb des PCIF-30 benötigten E/A-Ports verwendet.

Die Schalterpositionen sind von 1 bis 6 durchnumeriert. ***Stellen Sie mit dem Schalter 6 die obere Hexadezimalstelle ein. Mit den Schaltern 5, 4, 3 und 2 stellen Sie die mittlere Hexadezimalstelle ein und mit dem Schalter 1 die untere Hexadezimalstelle.***



Schaltereinstellung - Beispiel

In diesem Beispiel wird gezeigt, wie Sie die Moduladresse auf 310 einstellen.



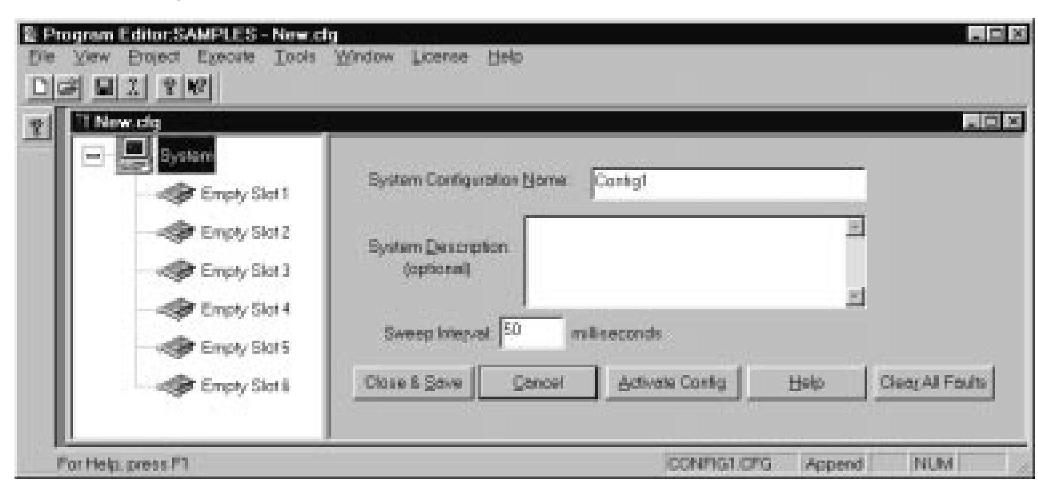
Konfiguration der PCIF-30-Karte

1. **Starten Sie die PC Control *Programmeditor* Software.**

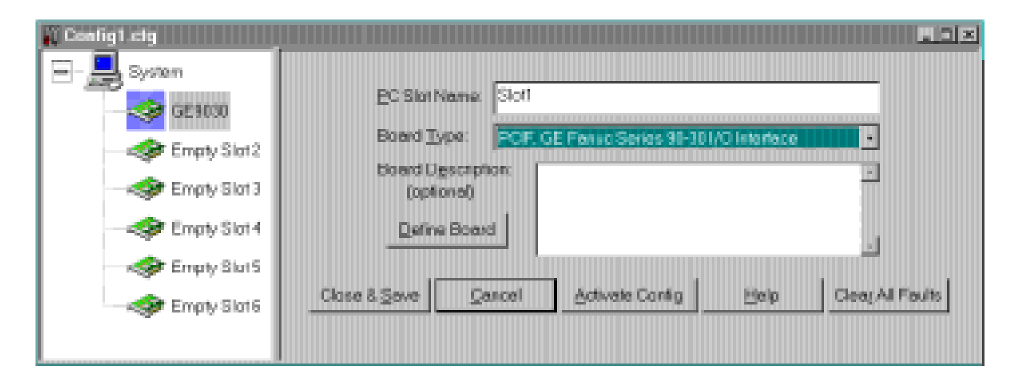
   Wählen Sie im Startmenü "PC Control Anwendungen", Programmeditor.

2. **Konfiguration der PCIF-30-Karte.**

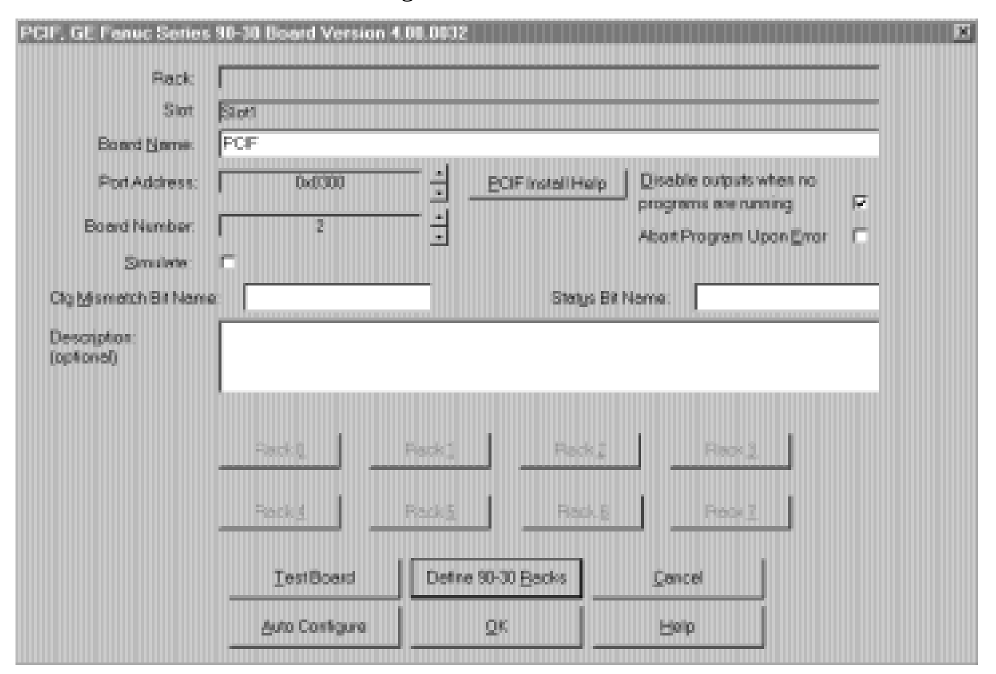
   A. Klicken Sie im PC Control Programmeditor-Dateimenü auf "Konfiguration öffnen" (oder "Neue Konfiguration"). Hierauf erscheint folgendes Konfigurationsfenster:



B. Wählen Sie aus der Baumanzeige auf der linken Seite des Fensters den Steckplatz aus, den Sie konfigurieren wollen. Hierauf erscheint folgendes Fenster:



C. Wählen Sie im Feld "Modultyp" die Option "PCIF, GE Fanuc Series 90-30 E/A-Schnittstelle" aus. Klicken Sie auf die Schaltfläche "Moduldefinition". Hierauf erscheint die Dialogbox "PCIF-Modul".



D. Stellen Sie die Konfigurationsparameter für die PCIF-30-Karte ein. Stellen Sie sicher, daß die Portadresse richtig eingestellt ist. Die Portadresse muß mit der Einstellung der DIP-Schalter übereinstimmen (siehe Seite B-3).

Weitere Einzelheiten zur Konfiguration finden Sie in der PC Control Online-Hilfe.

E. Nachdem Sie die Portadresse eingestellt haben, klicken Sie zur Überprüfung der Installation auf die Schaltfläche "Modultest". (Das Runtime-Subsystem darf nicht aktiv sein). Es sollte eine Dialogbox "Modultest" erscheinen, die die Meldung "PCIF Board found, shared RAM access successfully found” [PCIF-Modul gefunden, auf gemeinsam genutzten RAM erfolgreich zugegriffen] enthält. Erscheint diese Meldung nicht, sollten Sie versuchen, einen anderen Block von gemeinsam genutztem RAM zu konfigurieren.

3. Series 90-30 E/A konfigurieren.

Von der Dialogbox "PCIF-Modul" aus können Sie den Konfigurationsprozeß fortsetzen.

- **Automatische Konfiguration:** Sie können diese Funktion einsetzen, wenn E/A-Chassis und Module eingebaut und eingeschaltet sind und das Runtime-Subsystem nicht aktiv ist. Wenn Sie auf die Schaltfläche Automatische Konfiguration klicken, dann aktiviert die Software die PCIF-30-Karte und liest die Informationen über angeschlossene Chassis und Module.
- **Manuelle Konfiguration:** Klicken Sie auf die Schaltfläche "90-30 Chassis definieren", um diese Funktion zu benutzen. Hierauf erscheint die Dialogbox "Chassisdefinition".

Weitere Informationen finden Sie in der Online-Hilfe zur PC Control Software.

Anhang C — Personalcomputer-Schnittstelle (PCIF2) für Series 90-30 E/A

Übersicht

Über die IC693PIF400 Personalcomputer-Schnittstellenkarte kann die PC Control Software mit bis zu sieben Erweiterungschassis oder dezentralen Chassis der Series 90-30 kommunizieren. Die E/A-Chassis können dabei bis zu 210 m vom Personalcomputer entfernt sein. Die PC Control Software kann über das PIF400 bis zu 25.886 Bytes E/A überwachen und steuern. Die praktischen Grenzen der E/A-Kapazität ergeben sich aus den Zyklusanforderungen und der E/A-Konfiguration.

Die PIF400-Karte wird in einen 16-Bit-Steckplatz am PC/AT/ISA-Bus des Personalcomputers gesteckt. Erweiterungschassis und dezentrale Chassis werden an die Karte über eine 25-polige Buchse Typ "D" in einer verketteten Struktur angeschlossen. Die PIF400 liefert ein durch einen Überwachungszeitgeber überwachtes RUN-Ausgangssignal und ein Relais, die die Integration in Sicherheitsschaltkreise ermöglichen. Technische Daten finden Sie in dem mit dem Schnittstellenmodul mitgelieferten Datenblatt GFK-1540.

In diesem Anhang finden Sie Informationen zu folgenden Themen:

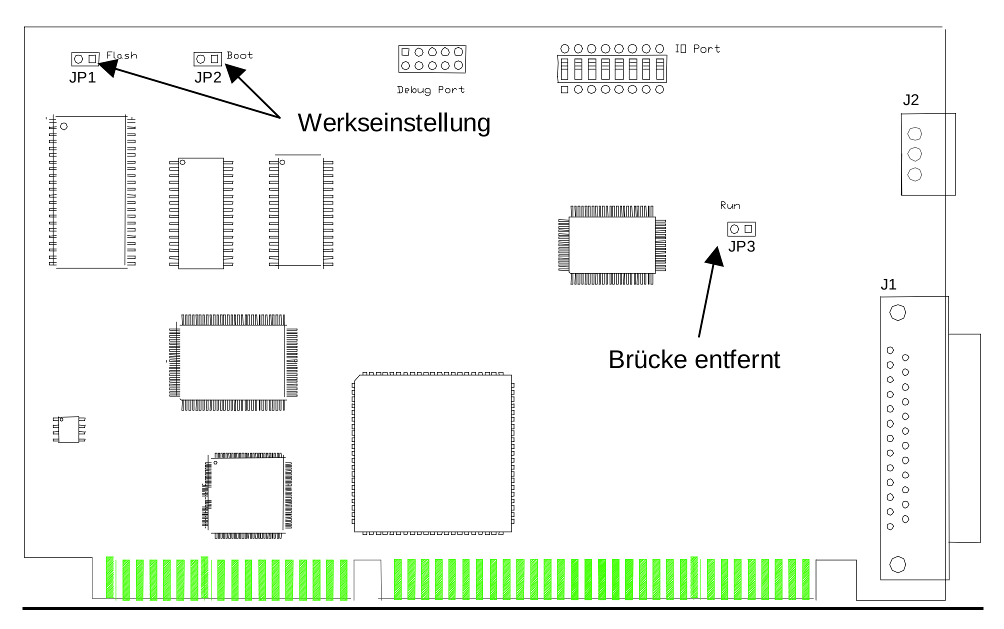
- Kompatibilität
- Hardwareübersicht
- Brücken
- Steckverbinder
- DIP-Schalter
- Schnellstart-Leitfaden

Kompatibilität

Die PIF400 unterstützt alle diskreten und analogen E/A-Module der Series 90-30, die von PC Control konfiguriert werden können.

Hardwareübersicht

Die PIF400-Karte besitzt eine konfigurierbare Brücken, einen E/A-Kabelsteckverbinder und ein RUN-Relais.



Brücken

JP3 – Anlauf beim Einschalten

Über diese Brücken kann festgelegt werden, ob die PIF400 vom PC zum Anlauf ein Startsignal benötigt.

Wird J3 entfernt, läuft die PIF400 erst an, wenn sie vom PC ein Startsignal empfängt. Wird der PC rückgesetzt, dann wird auch die PIF400 rückgesetzt und wartet auf das Startsignal vom PC.

Ist J3 eingesetzt, dann läuft die PIF400 hoch, sobald die Versorgungsspannung des PC eingeschaltet wird. Wird der PC rückgesetzt, dann wird die PIF400 *nicht* rückgesetzt und fährt mit dem Betrieb fort..

In der Standardposition ist die Brücke JP3 *entfernt*.

JP1 – Flash-Schutz und JP2 – Boot-Schutz

Diese Brücken werden im Werk eingestellt und dürfen nicht verändert werden.

Steckverbinder

J1 – Steckverbinder zu Erweiterungs-/dezentraler E/A

Über diese 25-polige Buchse vom Typ "D" erfolgt der Datenaustausch mit bis zu sieben E/A-Chassis. Diese Buchse wird mit dem Erweiterungsport des ersten E/A-Chassis im System verbunden.

Steckerbelegung des J1 Erweiterungs-Steckverbinder

Stift	Signalname	Beschreibung	Richtung
1	SHLD	Schirm	--
7	GND	Masse	--
2	DFRAME+	Datenrahmen-Signalpaar	Ausgang
3	DFRAME-		
8	RUN+	RUN-Signalpaar	Ausgang
9	RUN-		
12	PERR+	Paritätsfehler-Signalpaar	Eingang
13	PERR-		
16	DATA+	Daten-Signalpaar	Eingang und Ausgang
17	DATA-		
20	RSEL+	Busauswahl-Signalpaar	Ausgang
21	RSEL-		
24	IOCLK+	Datentakt-Signalpaar	Ausgang
25	IOCLK-		

Für den Anschluß der PIF400-Karte an die E/A-Chassis müssen Sie ein Kabel bereitstellen. Dieses Kabel muß aus sieben verdrillten Paaren mit gemeinsamer Abschirmung und Ableitungsader bestehen (Belden 8107 oder äquivalent). Das letzte Chassis am E/A-Bus muß mit einem E/A-Busabschlußstecker (Bestellnummer IC693ACC307) abgeschlossen werden.

Als Verbindungskabel können Erweiterungskabel von GE Fanuc in Standardlängen (siehe nachstehende Tabelle) oder selbstgefertigte Kabel verwendet werden. Dezentrale Chassis können bis zu 213 Meter vom PC entfernt sein, Erweiterungschassis bis zu 15 Meter.

Kabeltyp	Bestellnummer
1-Meter *T*-Kabel	IC693CBL300
2-Meter *T*-Kabel	IC693CBL301
15-Meter Punkt-zu-Punkt-Kabel	IC693CBL302

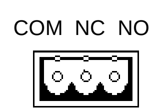
J2 – Anschluß RUN-Relais (Überwachungszeitgeber)

Wird das RUN-Signal zur Rückwandplatine der 90-30 aktiviert, dann wird das Relais durchgeschaltet, das diesen Steckverbinder ansteuert.

Das RUN-Relais kann an externe Sicherheitseinrichtungen angeschlossen werden, um einen Ausfall von Computer oder Anwendersoftware zu signalisieren. Im Normalbetrieb mit aktiver E/A wird der PIF400-Überwachungszeitgeber fortlaufend rückgesetzt und hält das Relais im durchgeschalteten Zustand. Führt die Anwendersoftware keine Aktualisierung der Ausgänge oder Zugriffe auf den RUN-Kontakt mehr durch, schaltet der Überwachungszeitgeber nach einer Sekunde alle Ausgänge ab. Dies bedeutet, daß alle Ausgangsmodulkreise in den Zustand AUS gehen, die RUN LEDs in den einzelnen Chassis-Stromversorgungen erlöschen und das Relais in seinen (abgeschalteten) Ruhezustand geht.

Steckerbelegung für J2, Anschluß RUN-Relais

Stift	Signalname
1	Masse
2	Öffnerkontakt (NC)
3	Schließerkontakt (NO)



RUN-Relais Kontaktspezifikationen

Anfangswiderstand	50 Milliohm
Maximale Schaltleistung	60 Watt, 62,5 VA
Maximale Schaltspannung	220 VDC, 250 VAC
Maximaler Schaltstrom	2 A
Maximale Strombelastbarkeit	3 A
UL/CSA Nennwerte	125 VAC @ 0,3 A 110 VDC @ 0,3 A 30 VDC @ 1,0 A
Mindestanzahl Schaltspiele Mechanisch Elektrisch	 100.000.000 500.000 (30 VDC @ 1,0 A, ohmsche Last) 100.000 (30 VDC @ 2,0 A, ohmsche Last)

P1 –Programmiergeräteport

Auf diesen Stiften dürfen ***keine*** Brücken gesteckt sein.

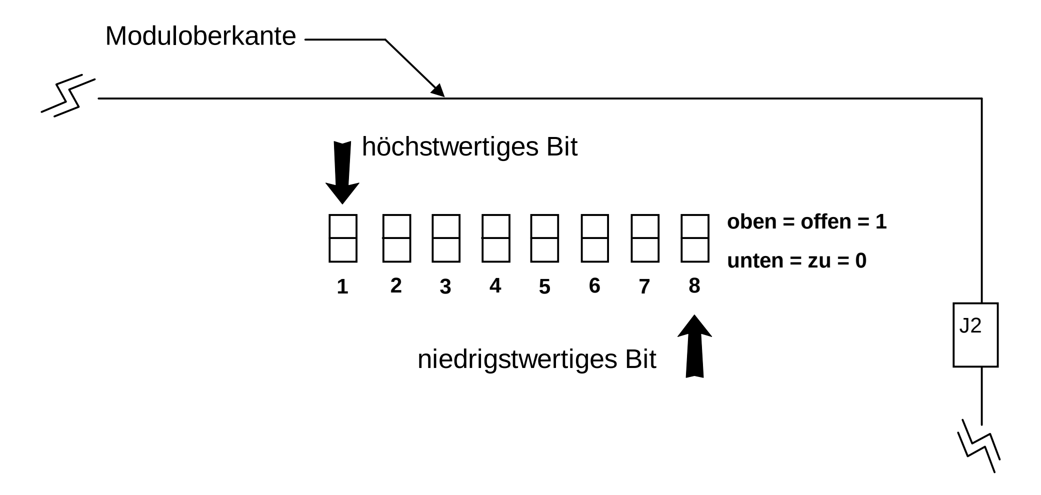
DIP-Schalter

S1 – E/A-Portadresse Auswahlschalter Mit diesem achtstelligen DIP-Schalter wählen Sie die Grundadresse im E/A-Adreßraum des PCs aus, unter der die PIF400 reagiert.

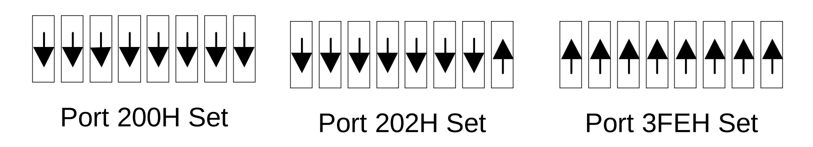
Die PIF400 benutzt zwei nebeneinanderliegende E/A-Ports, die auf einer beliebigen 2-Byte-Grenze zwischen 200 (hexa) und 3FE (hexa) liegen. Die Standardeinstellung ab Werk ist 200 (hexa). Die Einstellungen aller möglichen Adressen sind auf dem mit dem Schnittstellenmodul mitgelieferten Datenblatt aufgeführt.

Hinweis

S1 kann ein anderes Aussehen und/oder eine andere Markierung aufweisen. ***Die Schaltereinstellung basiert auf der nachstehend gezeigten physikalischen Schalterposition, nicht auf der eingravierten Numerierung.***



Lage der DIP-Schalter S1



Beispiele zur Einstellung der S1 Schalter

Schnellstart-Leitfaden

1. **Überprüfen Sie die Systembetriebsmittel.**

   Die PIF400-Karte benötigt im E/A-Portbereich Ihres Computers zwei benachbarte E/A-Ports. Zum Reservieren dieser Ports stellen Sie die Moduladresse mit dem DIP-Schalter S1 ein. Ehe Sie die PIF400-Karte in Ihren PC einbauen, sollten Sie festlegen, welche Ports Sie benutzen können, ohne dadurch mit E/A-Ports in Konflikt zu kommen, die den bereits in Ihrem Computer installierten Geräten zugewiesen wurden.

   Die PIF400-Karte benötigt 32 kB (7FFFH) im gemeinsam genutzten RAM. Die Anfangsadresse dieses Speichers wird von PC Control konfiguriert (Standard ist D0000). Sie müssen sicherstellen, daß dieser Speicher nicht von anderen Geräten verwendet wird.

   Um die benutzten Betriebsmittel zu ermitteln gehen Sie wie folgt vor:

   A. Melden Sie sich an unter einem Konto mit Systemadministrator-Privilegien.

   B. Wählen Sie im Windows NT® Startmenü die Optionen Programme, Administrative Werkzeuge, Windows NT-Diagnosefunktionen.

   C. Wählen Sie die Karteikarte "Betriebsmittel". Klicken Sie auf die Schaltfläche "E/A" und schauen Sie dann in der E/A-Port- Dialogbox nach unbenutzten Adressen.

   D. Suchen Sie einen Block mit zwei unbenutzten (nicht aufgelisteten) E/A-Portadressen für die PIF400. Der Schalter S1 muß auf den ersten Port in dem Block eingestellt werden. Die Standardeinstellung ab Werk für S1 ist 200 (hexa); werden die Ports 200—202 verwendet, dann müssen Sie die Einstellung von S1 verändern (siehe Schritt 2, "DIP-Schalter einstellen”).

   E. Klicken Sie auf die Karteikarte "Speicher", um die Speicherverfügbarkeit zu überprüfen. Suchen Sie einen freien Block von 32 kB und notieren Sie ihn zur Verwendung in der Softwarekonfiguration der PIF400-Karte. Wird die Standardeinstellung (D0000—D7FFF) verwendet, dann müssen Sie in PC Control einen anderen Speicherblock konfigurieren.

2. **Stellen Sie die DIP-Schalter ein.**

Hinweis

Sie brauchen die Einstellungen des DIP-Schalters S1 nur zu verändern, wenn Konflikte mit anderen im Computer eingebauten Schnittstellenmodulen auftreten. Die Standardeinstellung ab Werk ist 200 (hexa).

Einzelheiten zum Ändern der DIP-Schaltereinstellungen finden Sie auf Seite C-6.

3. Installieren Sie die PIF400-Karte in Ihrem PC.

Hinweis

Der PC und die Series 90 E/A-Chassis müssen mit einem gemeinsamen Masseanschluß verbunden werden. Normalerweise wird dieser gemeinsame Masseanschluß erreicht, indem PC und E/A-Chassis an die gleiche Spannungsquelle mit einem gemeinsamen Massebezugspunkt gelegt werden. Dies muß jedoch für jede Installation einzeln überprüft werden.

A. Bauen Sie die IC693PIF400 in einen leeren PC/AT/ISA-Erweiterungssteckplatz im PC ein. Befolgen Sie hierbei die Anweisungen des Computerherstellers. Um maximale Störfestigkeit zu erreichen muß zwischen den metallischen E/A-Klammern der PIF400-Karte und dem PC-Chassis ein guter elektrischer Kontakt bestehen (verwenden Sie die Schrauben, die beim Entfernen der Blindplatte frei wurden), und die Stromversorgung des Computers muß gut geerdet sein.

B. Schließen Sie die Series 90-30 Erweiterungschassis und/oder dezentrales Chassis an den 25-poligen "D" Erweiterungs-Steckverbinder (J1) der Karte an. Einzelheiten hierzu finden Sie auf Seite C-4.

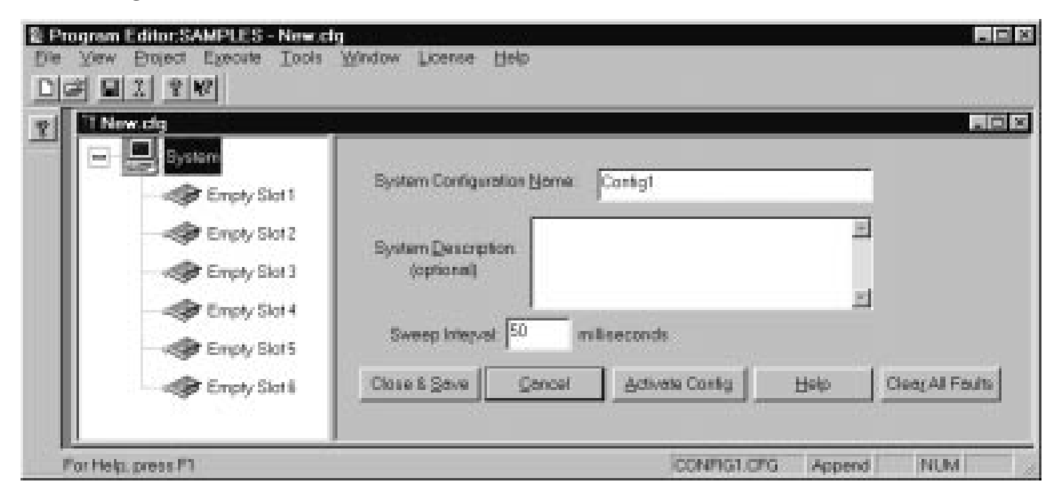
C. Schließen Sie den Steckverbinder des RUN-Relais (J2) zur Steuerung externer Geräte an. Achten Sie darauf, daß die angegebenen Kontaktbelastungen des Relais nicht überschritten werden. Einzelheiten hierzu finden Sie auf Seite C-5.

4. Starten Sie die PC Control *Programmeditor* Software.

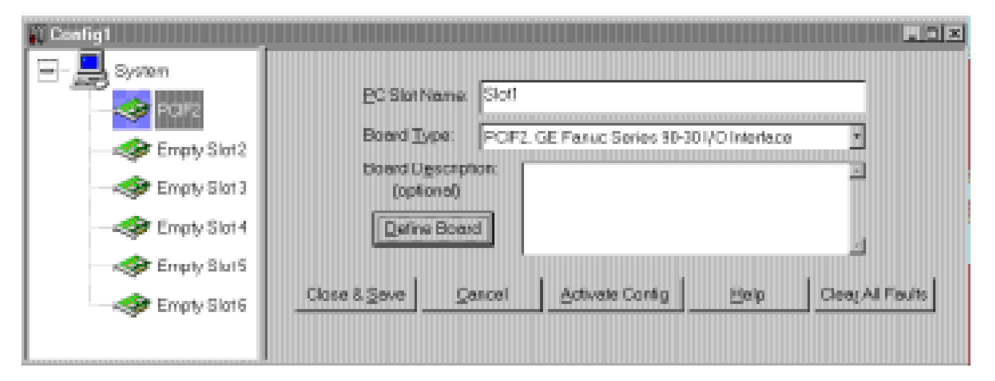
Wählen Sie im Startmenü "PC Control Anwendungen", Programmeditor.

5. Konfigurieren Sie die PIF400-Karte.

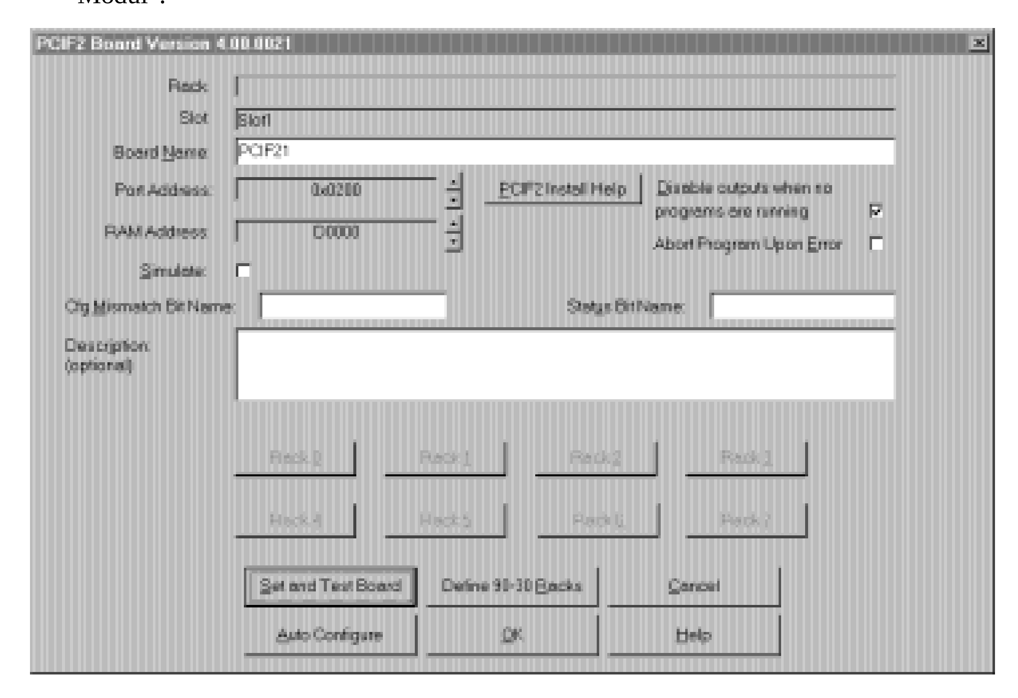
A. Klicken Sie im PC Control Programmeditor-Dateimenü auf "Konfiguration öffnen" (oder "Neue Konfiguration"). Hierauf erscheint folgendes Konfigurationsfenster:



B. Wählen Sie aus der Baumanzeige auf der linken Seite des Fensters den Steckplatz aus, den Sie konfigurieren wollen. Hierauf erscheint folgendes Fenster:



C. Wählen Sie im Feld "Modultyp" PCIF2 (für PIF400). Klicken Sie auf die Schaltfläche "Moduldefinition". Hierauf erscheint die Dialogbox "PCIF2-Modul".



D. Stellen Sie die Konfigurationsparameter für die PIF400 ein. Stellen Sie sicher, daß Portadresse und RAM-Adresse richtig eingestellt sind. (Siehe "Systembetriebsmittel überprüfen” auf Seite C-7.)

- Die Portadresse muß mit der Einstellung des DIP-Schalters S1 übereinstimmen.
- Für die RAM-Adresse müssen Sie einen freien Block von 32 kB auswählen. Weitere Einzelheiten zur Konfiguration finden Sie in der PC Control Online-Hilfe.

E. Nachdem Sie Portadresse und RAM-Adresse eingestellt haben, klicken Sie auf die Schaltfläche "Modul einstellen und testen" um zu überprüfen, ob die Installation in Ordnung ist. (Das Runtime-Subsystem darf nicht aktiv sein). Es sollte eine Dialogbox "Modultest" erscheinen, die die Meldung "PCIF2 Board found, shared RAM access successfully found” [PCIF2-Modul gefunden, auf gemeinsam genutzten RAM erfolgreich zugegriffen] enthält. Erscheint diese Meldung nicht, sollten Sie versuchen, einen anderen Block von gemeinsam genutztem RAM zu konfigurieren.

6. Konfigurieren Sie die Series 90-30 E/A.

Von der Dialogbox "PCIF2-Modul" aus können Sie den Konfigurationsprozeß fortsetzen.

- **Automatische Konfiguration:** Sie können diese Funktion einsetzen, wenn E/A-Chassis und Module eingebaut und eingeschaltet sind und das Runtime-Subsystem nicht aktiv ist. Wenn Sie auf die Schaltfläche Automatische Konfiguration klicken, dann aktiviert die Software die PCIF-Karte und liest die Informationen über angeschlossene Chassis und Module.
- **Manuelle Konfiguration:** Klicken Sie auf die Schaltfläche "90-30 Chassis definieren", um diese Funktion zu benutzen. Hierauf erscheint die Dialogbox "Chassisdefinition".

Weitere Informationen finden Sie in der Online-Hilfe zur PC Control Software.

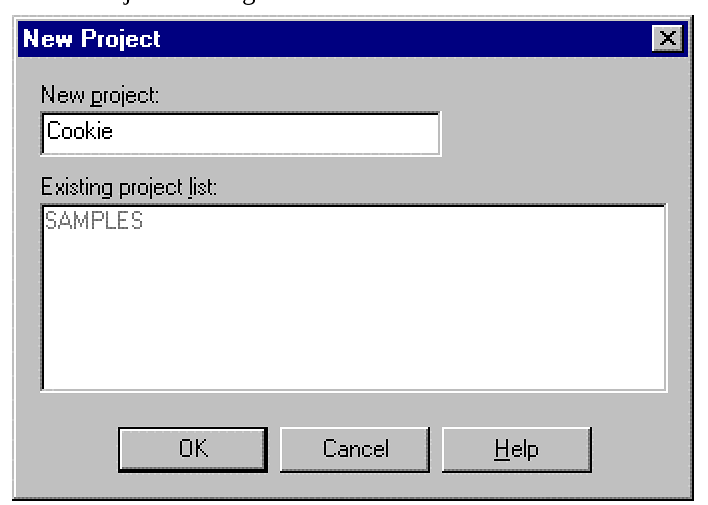
Anhang D

Anwendungsbeispiele

Dieser Anhang geht schrittweise durch ein Anwendungsbeispiel und zeigt Ihnen, wie Sie ein Projekt erstellen, E/A konfigurieren, ein Anwenderprogramm mit KOP, AS und strukturiertem Text erstellen, ein Bedienerschnittstellenmenü erstellen und die Anwendung ablaufen lassen und beobachten. Über das Programmeditor-Hilfemenü können Sie auf ein zusätzliches Tutorial zurückgreifen, das Sie durch die Erstellung einer Ablaufketten-Anwendung führt.

Übung 1: Ein neues Projekt erstellen

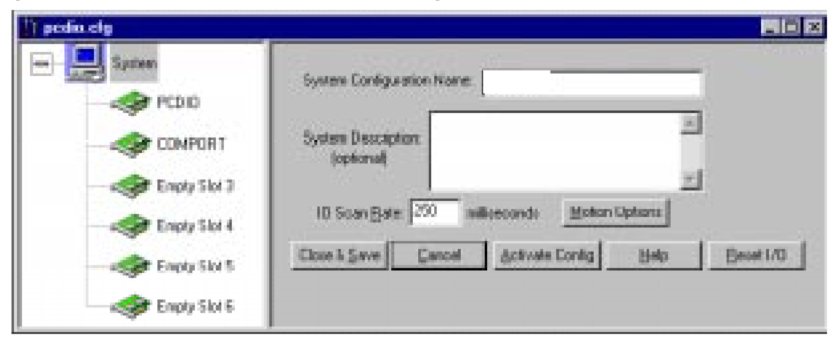
- Klicken Sie im Programmeditor auf das Menü **Projekt** und wählen dann **Neu**.
- Geben Sie in das Feld "Neues Projekt" Cookie ein. Ihr Bildschirm sollte so aussehen wie im nachstehenden Bild. Klicken Sie auf die Schaltfläche "OK", um das neue Projekt anzulegen.



Übung 2: Eine neue Systemkonfiguration erstellen

Systemdialog

- Wählen Sie im Programmeditor das Menü **Datei** und klicken dann auf **Neue Konfiguration**. Hierauf wird eine neue Systemkonfigurations-Dialogbox geöffnet (siehe nachstehende Abbildung).



- Verwenden Sie in diesem Beispiel den Standardnamen der Systemkonfiguration "Config1" sowie eine E/A-Zykluszeit von 250 ms.

Hinweis

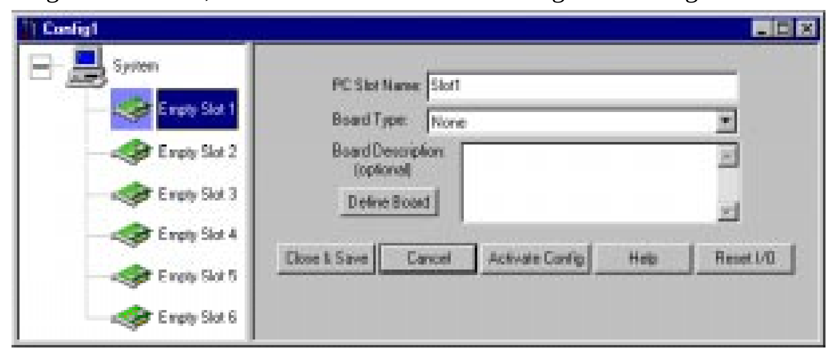
Die E/A-Zykluszeit gibt die Frequenz an, mit der PC CONTROL die E/A liest und aktualisiert und die Logik auswertet. Die PC CONTROL Zyklusrate ist nicht synchron mit der Zyklusrate der E/A-Hardware.

Steckplatzdialog

Der nächste Schritt bei der Konfiguration der E/A besteht darin, die lokalen E/A-Module, E/A-Scannermodule und/oder Bewegungsmodule zu definieren, die Sie in Ihrem System benutzen werden.

Der Steckplatzdialog enthält Angaben zur Art der Kommunikationskarte, die in einem bestimmten Steckplatz eingebaut wurde. Bei einem PC ist die eigentliche Steckplatznummer zwar ohne Bedeutung, Sie können aber durch die Numerierung einen Wartungstechniker bei Reparatur und Anschluß auf die entsprechende Karte hinweisen.

- Zur Konfiguration eines E/A-Moduls klicken Sie in der Graphik auf der linken Seite der Dialogbox auf **Leerer Steckplatz 1**. Nachdem ein Steckplatz ausgewählt wurde, sieht die rechte Seite der Dialogbox wie folgt aus:

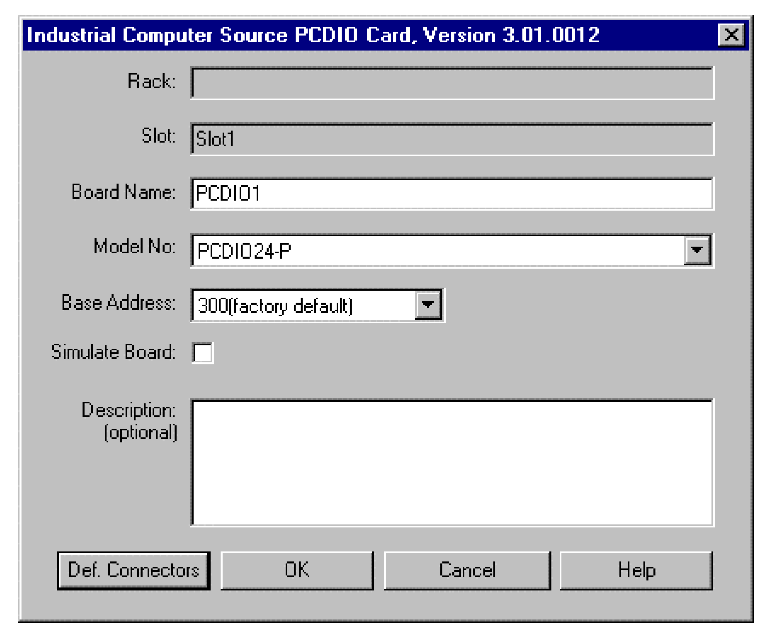


- Die neue Dialogbox enthält ein Dropdown-Menü mit der Bezeichnung "Modultyp". Die Dropdown-Liste enthält alle im System installierten PC CONTROL E/A-Treiber.
- Wählen Sie aus der Dropdown-Liste den Gerätetreiber "**Industrial Computer Source PCDIO**". Erscheint dieser Treiber nicht in Ihrer Liste, dann müssen Sie die PC CONTROL Software verlassen und den Gerätetreiber mit den Installationsdisketten für E/A-Treiber installieren
- Nachdem Sie das Modul ausgewählt haben, klicken Sie auf die Schaltfläche **Moduldefinition**, um die Software für die Kommunikation mit dem Modul zu konfigurieren.

Modul definieren

Die Moduldefinitions-Dialogbox ist für jedes E/A-Modul anders. Mit dem Dialog können Sie die PC CONTROL Software so konfigurieren, daß sie mit dem ausgewählten E/A-Modul Daten austauschen kann.

- Verwenden Sie in diesem Beispiel die in der Dialogbox angebotene Standardinformation (siehe nachstehende Abbildung). Besitzen Sie einen E/A-Simulator, müssen Sie ihn so konfigurieren, daß er mit diesen Standardwerten arbeitet. Haben Sie in Ihrem System keinen E/A-Simulator installiert, dann müssen Sie das Modul simulieren, indem Sie auf das Kontrollkästchen **Modul simulieren** klicken.



- Klicken Sie auf die Schaltfläche **Steckverbinder definieren**, um den E/A-Punkten Symbolnamen zuzuweisen.

Steckverbinder definieren

- Eine Dialogbox "Portdefinition" wird angezeigt, wenn Sie auf die Schaltfläche **Steckverbinder definieren** klicken.
- Wählen Sie für den ersten Port **Eingang** und geben dann die Symbolnamen "switch1" bis "switch6" für die ersten sechs Punkte ein. Diese Symbole entsprechen den sechs Schaltern des E/A-Simulators.
- Wählen Sie für den zweiten Port **Ausgang** und geben dann die Symbolnamen "light1" bis "light6" für die ersten sechs Punkte und "buzzer" für den siebten

Punkt ein. Diese Symbole entsprechen den sechs Lampen und dem Summer des E/A-Simulators. Für "Name Port B" geben Sie "lights" ein.

Hinweis

Bei den Symbolnamen wird in PC Control zwischen Groß- und Kleinbuchstaben unterschieden. In diesem Beispiel verwenden wir nur Kleinbuchstaben für die Symbolnamen.

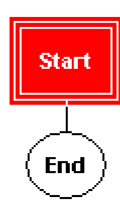
Die neue Konfiguration speichern und aktivieren

- Klicken Sie auf die Schaltfläche **OK** im Portdefinitionsdialog. Klicken Sie dann auf **Ja**, wenn Sie gefragt werden "Wollen Sie dies Konfiguration aktivieren?"
- Klicken Sie auf **Ja**, wenn Sie gefragt werden, ob Sie vor der Aktivierung speichern wollen.
- Die Dialogbox **Speichern als** wird geöffnet und gestattet Ihnen, der Konfiguration einen Namen zuzuweisen. Geben Sie "**pcdio**" in das Feld "Dateinamen" ein und klicken dann auf **OK**.
- Nachdem die neue Konfiguration aktiviert wurde, erscheint erneut die Dialogbox "Systemkonfiguration". Klicken Sie auf **Schließen und Speichern**, um die Erstellung der Konfiguration zu beenden.

Übung 3: Eine Ablaufkette erstellen

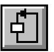
Läuft der Programmeditor noch nicht, dann öffnen Sie die PC Control Anwenderprogrammgruppe und klicken auf das Symbol Programmeditor, um das Dienstprogramm zu starten.

- Drücken Sie in der Editor-Symbolleiste oben im Menü die Schaltfläche "Neue Datei", um eine neue Programmdatei anzulegen. Wählen Sie aus der Dialogbox AS+/M und klicken dann auf OK, um eine neue Ablaufkette. Hierauf wird ein Fenster geöffnet, das die neue Ablaufkette enthält.

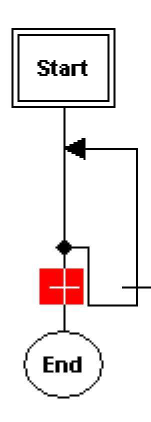


Programmstruktur erstellen

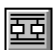
- Wählen Sie das Schleifenwerkzeug aus der AS-Symbolleiste.



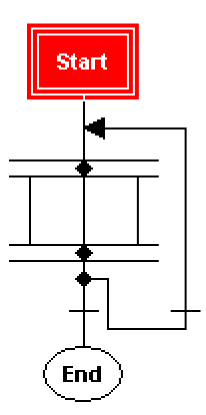
- Setzen Sie den Cursor auf den Steuerpfad zwischen dem Startschritt und dem Endeschritt. Drücken Sie die linke Maustaste, um ein Steuerschleifenelement in der Ablaufkette abzulegen.



- Wählen Sie das Simultanverzweigungs-Werkzeug aus der AS-Symbolleiste.



- Setzen Sie den Cursor auf die Mitte der Schleife und drücken die linke Maustaste, um ein Simultanverzweigungselement in der Ablaufkette abzulegen.
- Wählen Sie erneut das Simultanverzweigungs-Werkzeug und fügen der Verzweigung einen dritten Zweig hinzu, indem Sie auf die obere waagerechte Doppellinie klicken.



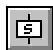
Richtlinien für den Gebrauch von Simultanverzweigungen

Beachten Sie die folgenden Richtlinien, wenn Sie eine Simultanverzweigung erstellen.

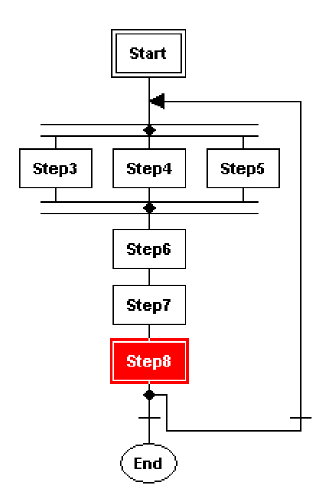
- Sprechen Sie die gleiche Variable nicht in unterschiedlichen Pfaden einer Simultanverzweigung an.
- Rufen Sie die gleiche Tochter-Ablaufkette nicht aus Makroschritten in unterschiedlichen Pfaden einer Simultanverzweigung auf.
- Um eine saubere Verzweigung sicherzustellen, dürfen Sie Marken nicht in folgender Weise verwenden:
  - um auf einen Punkt außerhalb einer Simultanverzweigung zu springen.
  - um in eine Simultanverzweigung hinein zu springen.
  - um zu einem anderen Pfad innerhalb einer Simultanverzweigung zu springen.

Schritte hinzufügen

Wählen Sie das Schrittwerkzeug aus der AS-Symbolleiste.

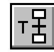


Setzen Sie den Cursor auf den linken Steuerpfad der Simultanverzweigung. Drücken Sie die linke Maustaste, um ein Schrittelement in der Ablaufkette abzulegen. Setzen Sie den Cursor auf die nächsten beiden Steuerpfade und legen Sie in jedem ein Schrittelement ab. Setzen Sie den Cursor auf den Steuerpfad unterhalb der Verzweigung und oberhalb des Schleifenwerkzeugs. Fügen Sie an diesem Punkt drei aufeinanderfolgende Schrittelemente in die Ablaufkette ein.

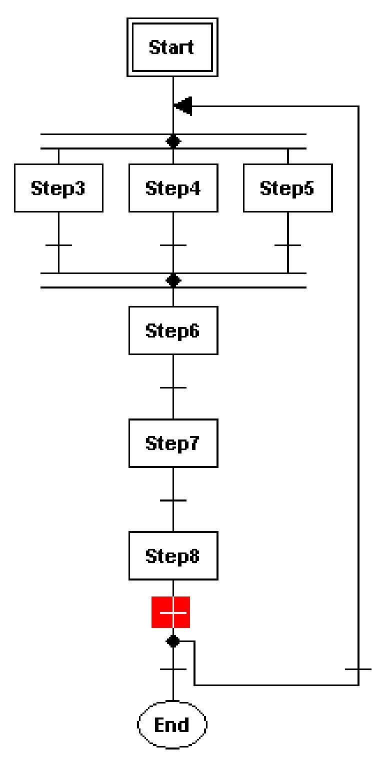


Transitionen hinzufügen

Wählen Sie das Transitionswerkzeug aus der AS-Symbolleiste.



- Setzen Sie den Cursor unter die einzelnen Steuerpfade in der Simultanverzweigung und klicken Sie die linke Maustaste, um ein Transitionselement in der Ablaufkette abzulegen. Wiederholen Sie diesen Vorgang nach jedem der drei Schritte außerhalb der Simultanverzweigung.



Anwenderprogramm speichern



Speichern Sie Ihre bisherige Arbeit, ehe Sie Einzelheiten zu Ihrer Ablaufkette hinzufügen. Verwenden Sie den Menübefehl "Datei\Speichern" oder die Schaltfläche "Datei speichern" auf der Editor-Symbolleiste, um die aktive Datei unter ihrem aktuellen Namen und Verzeichnis zu speichern. Da Sie diese Datei zum ersten Mal speichern, zeigt der Programmeditor die Dialogbox "Speichern als" an, in der Sie der Datei einen Namen zuweisen können. Geben Sie "cookie_maker" im Feld "Dateinamen" ein und klicken dann auf die Schaltfläche **Speichern**. Die Kopfleiste der Ablaufkette muß wie folgt aussehen:



Übung 4: Symbole erstellen

Greifen Sie, mit dem neuen Programm "cookie_maker.SCF" als aktivem Fenster, auf den Symbolmanager zu, indem Sie die Schaltfläche "Symbolmanager" in der Editor-Symbolleiste drücken..



Der Symbolmanager zeigt die von Ihnen im Konfigurations-Dienstprogramm definierten E/A-Symbolnamen an. Da Sie aus jedem Anwenderprogramm eines Projektes heraus auf ein E/A-Symbol zugreifen können, steht vor den E/A-Symbolen ein rotes "G" (globales Symbol).

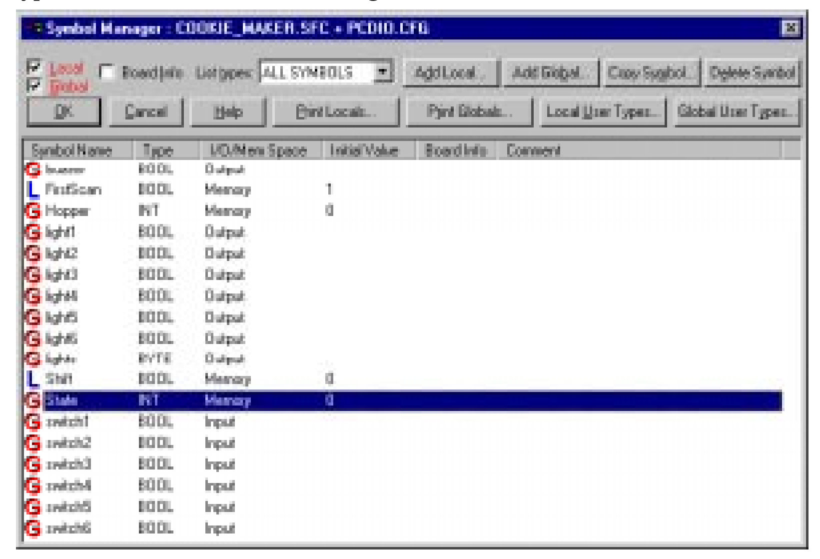
Lokale Symbole hinzufügen

- Klicken Sie auf die Schaltfläche **Lokal hinzu** oben im Symbolmanager. Geben Sie in der Dialogbox "Symboldetail" den Symbolnamen "**Erster Zyklus**" ein. Das Listenfeld "Typ:" muß dabei auf **BOOL** eingestellt sein. Geben Sie "**1**" als Anfangswert ein. Klicken Sie auf **OK**, um den Vorgang zu beenden und das neue Symbol zur Symbolmanagerliste hinzuzufügen.
- Fügen Sie ein weiteres lokales Symbol hinzu. Name: "**Shift**", Typ: **BOOL**, Anfangswert: **0**. Klicken Sie am Ende der Eingabe auf **OK**.

Globale Symbole hinzufügen

- Globale Symbole werden auf die gleiche Weise wie lokale Symbole erstellt. Klicken Sie auf die Schaltfläche **Global hinzu** oben im Symbolmanager.

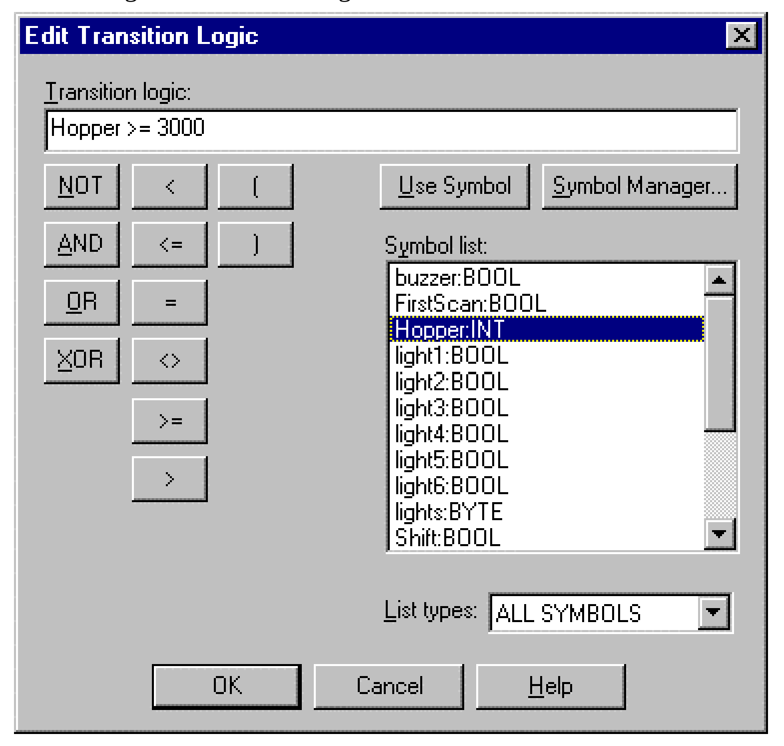
- Erstellen Sie zwei neue globale Symbole. Das erste heißt "**Hopper**", ist vom Typ **INT** und besitzt einen Anfangswert **0**. Das zweite heißt "**State**", ist vom Typ **INT** und besitzt einen Anfangswert **0**.



- Klicken Sie oben im Symbolmanager auf **OK**, um das Fenster zu schließen und die neuen Symbole zu akzeptieren. Da Sie neue globale Symbole hinzugefügt haben, fragt der Programmeditor, was Sie mit der globalen Information anfangen wollen. Klicken Sie auf die Schaltfläche **Speichern & Aktivieren**, um die globalen Symbole in der Systemkonfiguration zu speichern und die neue Konfiguration zu aktivieren.

Übung 5: Transitionslogik hinzufügen

- Doppelklicken Sie auf die Transition ganz links in der Simultanverzweigung, um die Dialogbox "Transitionslogik bearbeiten" zu öffnen.



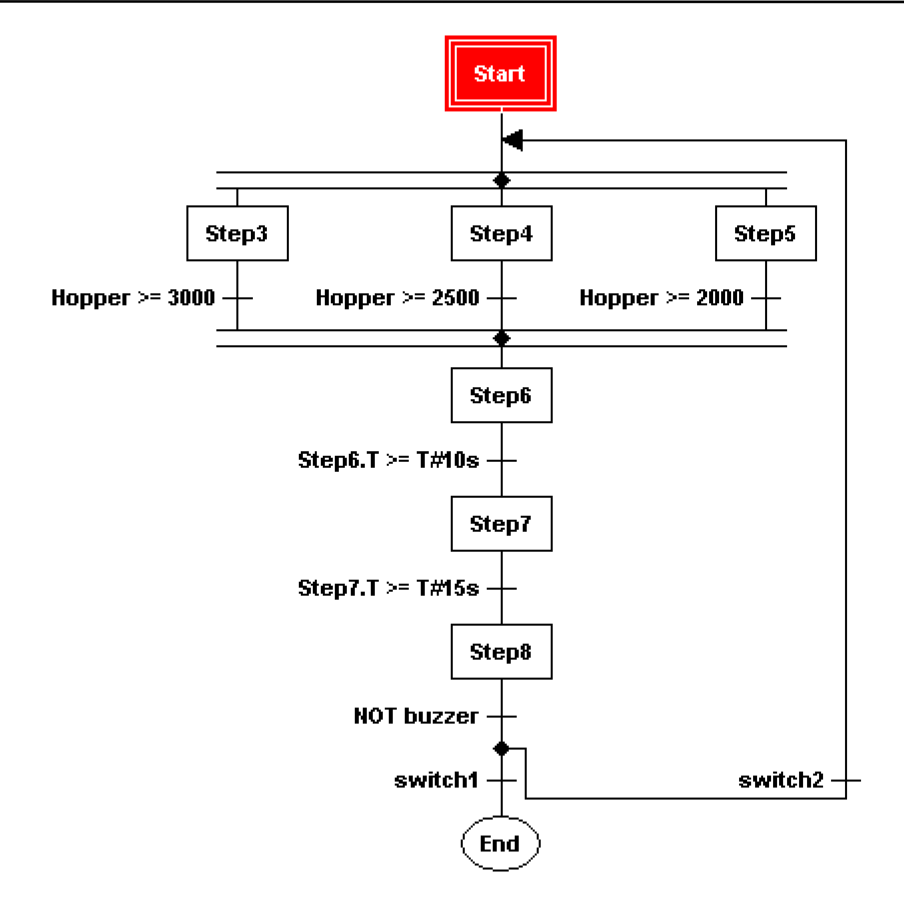
- Klicken Sie in der Symbolliste auf das Symbol **Hopper**, um es in die Dialogbox "Transitionslogik bearbeiten" zu kopieren. Klicken Sie dann auf die Schaltfläche **">="** und geben anschließend "**3000**" ein. Klicken Sie auf **OK**, um die Dialogbox "Transitionslogik bearbeiten" zu schließen und die Änderungen zu übernehmen.
- Wiederholen Sie diese Schritte für die nächsten beiden Pfade in der Simultanverzweigung. Geben Sie als Testbedingung für diese Transitionen "**2500**" bzw. "**2000**" ein.
- Doppelklicken Sie unmittelbar unter dem ersten Schritt nach der Verzweigung auf die Transition. Für diese Transitionslogik verwenden Sie eine der Schrittvariablen, die von PC CONTROL automatisch erstellt werden. Geben Sie in der Dialogbox "Transitionslogik bearbeiten" den Namen des Schritts ein, der unmittelbar darüber steht, und fügen am Ende ein "**.T**" hinzu (in diesem Beispiel: **Step6.T**). Klicken Sie dann auf die Schaltfläche **">="** und geben dann "**T#10s**" ein (die IEC-1131-Syntax für zehn Sekunden). Diese Transition wird also TRUE, wenn Step6 länger als 10 Sekunden aktiv ist.

- Wiederholen Sie diese Schritte für die Transition unterhalb von **Step7**. Verwenden Sie **T#15s**, um die Transition nach 15 Sekunden auf TRUE zu setzen.
- Bearbeiten Sie als nächstes die Transition unterhalb von **Step8**. Klicken Sie zunächst auf die Schaltfläche "**NOT**" und doppelklicken Sie dann in der Symbolliste auf das Symbol "**buzzer**". Klicken Sie auf "**OK**", um diese Änderungen zu übernehmen.

Zum Schluß bearbeiten wir noch die beiden Transitionsbedingungen auf dem **Schleifenwerkzeug**.

- Doppelklicken Sie auf die mit dem Schritt **End** verbundenen Transition. Blättern Sie die Symbolliste nach unten, bis das Symbol **switch1** sichtbar wird. Doppelklicken Sie auf das Symbol **switch1**, um es in die Dialogbox "Transitionslogik bearbeiten" zu kopieren. Klicken Sie auf **OK**, um die Dialogbox "Transitionslogik bearbeiten" zu schließen und die Änderungen zu übernehmen.
- Doppelklicken Sie auf die Transition im Rückkopplungskreis der Schleife. Blättern Sie die Symbolliste nach unten, bis das Symbol **switch2** sichtbar wird. Doppelklicken Sie auf das Symbol **switch2**, um es in die Dialogbox "Transitionslogik bearbeiten" zu kopieren. Klicken Sie auf **OK**, um die Dialogbox "Transitionslogik bearbeiten" zu schließen und die Änderungen zu übernehmen.

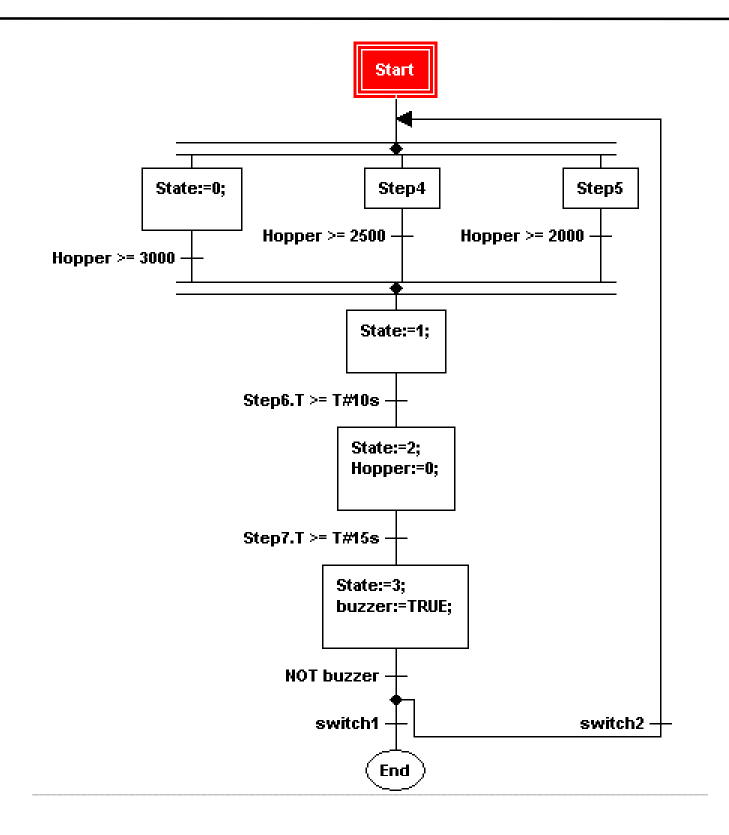
Nachdem Sie die Einzelheiten der Booleschen Transition eingetragen haben, sollte das Programm wie folgt aussehen:


Start
Step3
Step4
Step5
Hopper >= 3000
Hopper >= 2500
Hopper >= 2000
Step6
Step6.T >= T#10s
Step7
Step7.T >= T#15s
Step8
NOT buzzer
switch1
switch2
End

Übung 6: Eingabe von strukturierten Textbefehlen

- Doppelklicken Sie auf **Step3**, um die Dialogbox **Schritt bearbeiten** zu öffnen
- Wählen Sie als Befehlstyp **Strukturierter Text** aus. Wählen Sie die Dialogbox "Befehlsliste bearbeiten" und geben Sie folgenden Zuweisungsbefehl in strukturiertem Text ein:
  ```
  State := 0;
  ```
- Klicken Sie auf **OK**, um die Dialogbox "Schrittbearbeitung" zu schließen und die Änderungen zu übernehmen.
- Doppelklicken Sie auf **Step6**, um die Dialogbox **Schrittbearbeitung** zu öffnen, und geben Sie folgenden Befehl ein:
  ```
  State := 1;
  ```
- Doppelklicken Sie auf **Step7**, um die Dialogbox **Schrittbearbeitung** zu öffnen, und geben Sie folgenden Befehl ein:
  ```
  State := 2;
  Hopper := 0;
  ```
- Doppelklicken Sie auf **Step8**, um die Dialogbox **Schrittbearbeitung** zu öffnen, und geben Sie folgenden Befehl ein:
  ```
  State := 3;
  buzzer := TRUE;
  ```
- Um alle Schrittbefehle anzuzeigen, klicken Sie auf das Dropdown-Menü **Anzeigen** Dropdown-Menü und wählen dort **Alle Schritte/Als Befehle**.

Am Ende sollte Ihr Programm wie folgt aussehen:



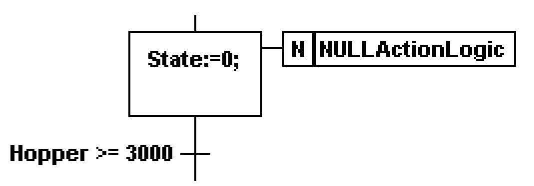
Übung 7: Aktionsblöcke hinzufügen und bearbeiten

Schritte 3,4 und 5 bearbeiten

- Wählen Sie das Aktionswerkzeug aus der AS-Symbolleiste.



- Setzen Sie den Cursor auf den Schritt ganz links (**Step3**). Klicken Sie, um diesen Schritt mit einer Aktion zu verbinden. Der Schritt sieht nun wie folgt aus:



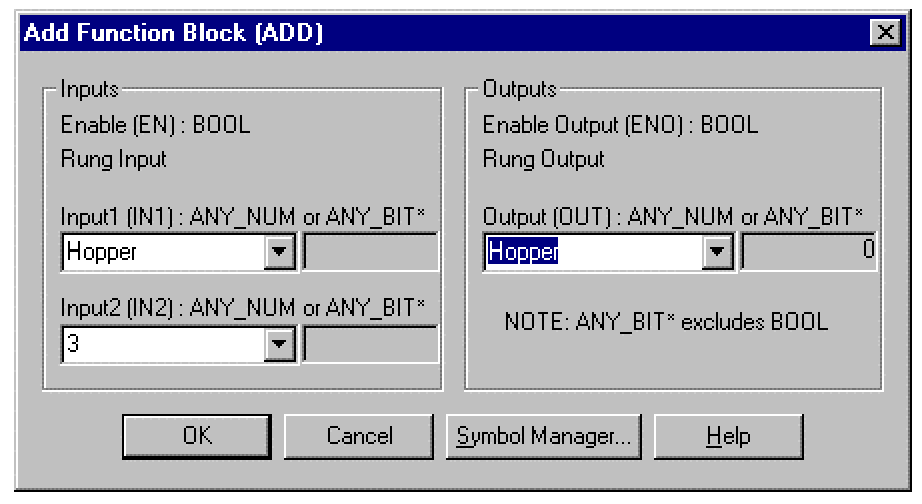
- Doppelklicken Sie auf die Aktionsbezeichnung, um die Dialogbox **Aktionsverknüpfung bearbeiten** zu öffnen. Wählen Sie die Editierbox "Aktionsbezeichnung". Geben Sie den Namen für die neue Aktion ein. Geben

Sie in unserem Beispiel den Namen "**Flour**" ein und klicken dann auf **OK**. Da diese Aktion noch nicht existiert, erscheint eine Dialogbox und fragt Sie, ob Sie sie anlegen wollen. Klicken Sie auf die Schaltfläche **OK**, um die neue Aktion anzulegen. Ein Fenster geht auf, das ein leeres Kontaktplanprogramm für die neue Aktion enthält.

- Wählen Sie über das Drop-Down-Listenfeld auf der Funktionsblockpalette die Palette **Arithmetische Funktionen**.



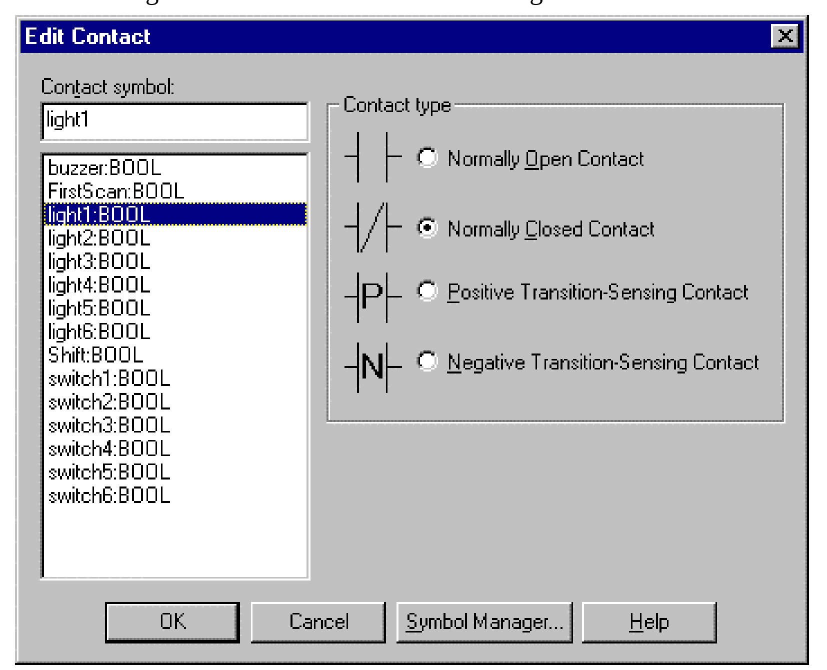
- Klicken Sie auf die Funktionsblock-Schaltfläche **ADD**. Der Cursor wird zum Funktionsblock-Cursor. Setzen Sie den Cursor auf den horizontalen Strompfad des Kontaktplanfensters. Klicken Sie die linke Maustaste, um einen Funktionsblock "ADD" abzulegen.
- Hierauf erscheint die Dialogbox "Funktionsblock hinzufügen (ADD)". Geben Sie **Hopper** in die Editierbox **Input1** ein. Geben Sie **3** in die Editierbox **Input2** ein. Geben Sie **Hopper** in die Editierbox **Output** ein. Solange die Aktion aktiv ist, wird hierdurch in jedem E/A-Zyklus der Wert "3" zu der Variablen "Hopper" addiert.



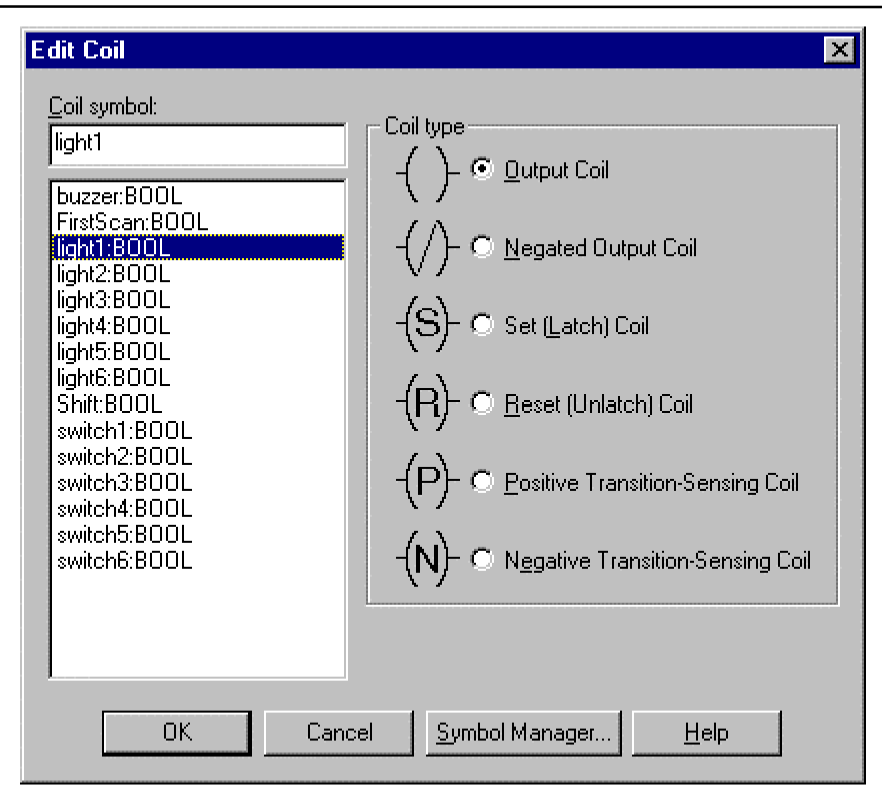
- Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.
- Klicken Sie auf die Schaltfläche **Fenster schließen**, um das Aktionsblock-Kontaktplanprogramm zu schließen.
- Benennen Sie die Aktionsblöcke in den Schritten 4 und 5: **Sugar** bzw. **Chips**. Plazieren Sie in der Aktion "Sugar" einen Funktionsblock **ADD**, der zu **Hopper** eine "4" addiert, und in der Aktion "Chips" einen Funktionsblock **ADD** , der zu **Hopper** eine "5" addiert.

Step6 bearbeiten

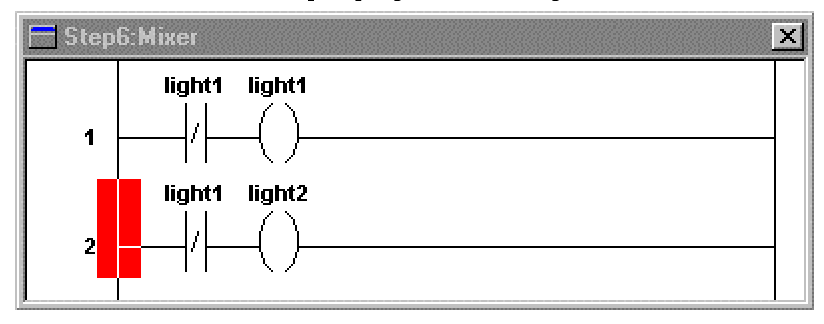
- Benennen Sie den mit **Step6**, **Mixer**, verbundenen Aktionsblock.
- Wählen Sie das **Strompfadwerkzeug** aus der Kontaktplan-Symbolleiste. Der Cursor wird zum neuen Strompfad-Cursor. Setzen Sie den Cursor unmittelbar unter den bestehenden Strompfad und klicken Sie auf die linke Maustaste, um einen neuen Strompfad einzufügen.
- Wählen Sie das **Kontaktwerkzeug** aus der Kontaktplan-Symbolleiste. Der Cursor wird zum Kontakt-Cursor. Setzen Sie den Cursor auf den ersten Strompfad und klicken Sie die linke Maustaste, um einen Kontakt einzufügen.
- Hierauf erscheint die Dialogbox "Kontakt editieren". Wählen Sie **light1** aus der Symbolliste aus. Wählen Sie **Öffner** als Kontakttyp aus. Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.



- Wählen Sie das **Spulenwerkzeug** aus der Kontaktplan-Symbolleiste. Der Cursor wird zum Spulen-Cursor. Setzen Sie den Cursor auf den ersten Strompfad zur Rechten des Kontakts und klicken Sie die linke Maustaste, um eine Spule einzufügen.
- Hierauf erscheint die Dialogbox "Spule editieren". Wählen Sie **light1** aus der Symbolliste aus. Wählen Sie **Ausgangsspule** als Spulentyp. Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.

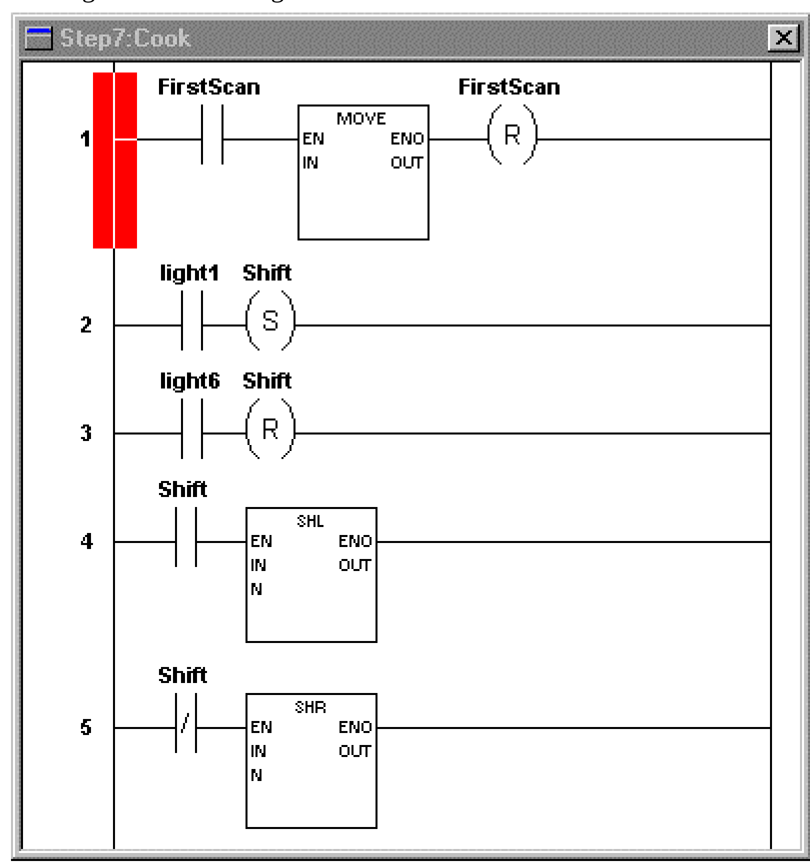


- Fügen Sie einen zweiten **light1, Öffner** in den zweiten Strompfad ein sowie eine **Ausgangsspule** für **light2**.
- Am Ende sollte Ihr Kontaktplanprogramm wie folgt aussehen:

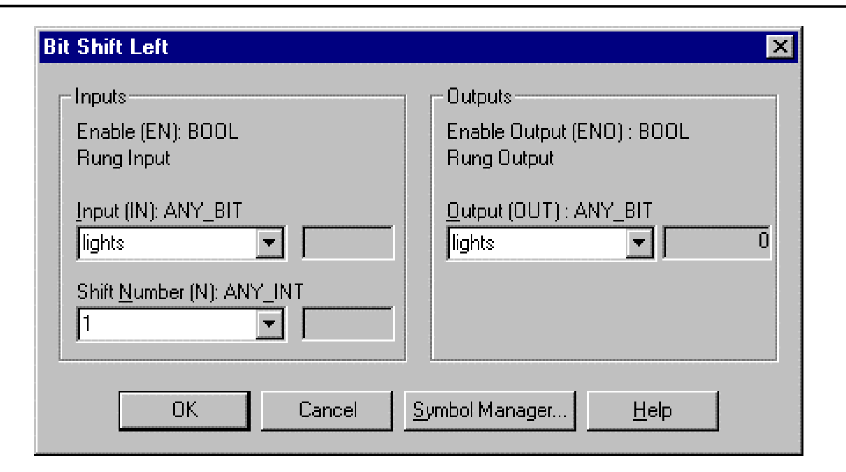


Step7

- Benennen Sie den mit **Step7**, **Cook**, verbundenen Aktionsblock.
- Fügen Sie Strompfade, Kontakte, Spulen und Funktionsblöcke hinzu, so daß sich folgende Struktur ergibt:

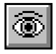


- Der Funktionsblock **MOVE** in Strompfad 1 schiebt eine "1" in das Symbol **lights.**
- Die Blöcke **Bit nach links schieben (BSL)** und **Bit nach rechts schieben (BSR)** in den Strompfaden 4 und 5 (die Funktionsblöcke zum Verschieben von Bits finden Sie in der Funktionsblockpalette **Bitweise**) verschieben das Symbol **lights** jeweils um **1**. Jede der Dialogboxen zum Verschieben von Bits muß wie folgt konfiguriert werden:

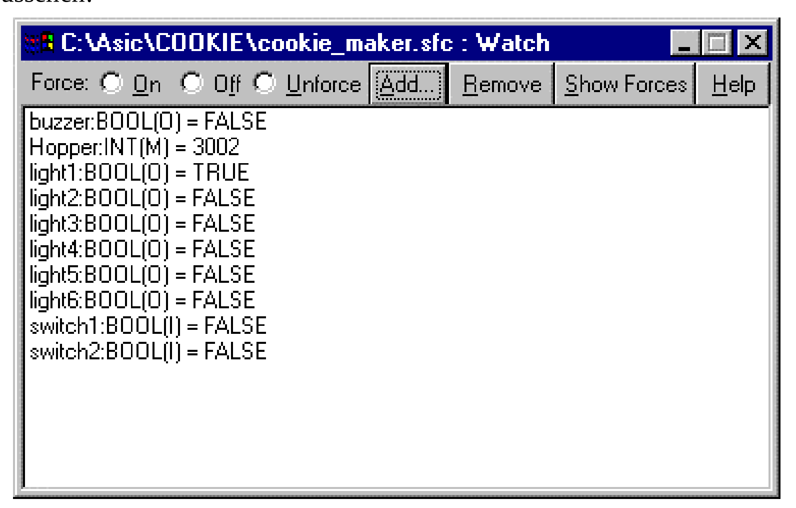


Übung 8: Beispielprogamm ausführen

- Verwenden Sie den Menübefehl "Ausführen\RUN", um das Programm cookie_makre.SFC ablaufen zu lassen. Sind die PC CONTROL Runtime-Subsysteme nicht aktiv, wird der Benutzer aufgefordert, die Runtime-Subsysteme zu starten. Läuft das Programm, ist die Programmanzeige hervorgehoben. Aktive Schritte und Transitionen werden in grün hervorgehoben.
- Öffnen Sie das **Überwachungsfenster**. Klicken Sie hierzu auf das Symbol Überwachungsfenster in der Programmeditor-Symbolleiste.



- Klicken Sie auf die Schaltfläche **Hinzufügen**. Selektieren Sie die folgenden Symbolnamen, indem Sie mit der linke Maustaste auf sie klicken:
  Klicken Sie auf **OK**, um die Dialogbox "Hinzufügen" zu schließen und die Symbole hinzuzufügen. Das Überwachungsfenster sollte nun wie folgt aussehen:



- Fixieren Sie die Symbole, wie es für die Steuerung des Programmablaufs erforderlich ist.

Übung 9: Ein Bedienerschnittstellenmenü erstellen

Zum Erstellen eines neuen Bedienermenüs wählen Sie den Befehl "Bearbeiten\Neues Menü aus dem Dropdown-Menü. Hierauf erscheint die Dialogbox "Neuen Menünamen auswählen". Geben Sie "**CookieMaker**" ein und drücken dann **OK**. Ein leeres Bedienermenü wird angelegt, bei dem der Menüname in der Kopfleiste steht.

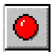


Bei der Erstellung einer Bedienerschnittstellendatei wird ein PC Control Standardmenü erzeugt und als das "Startmenü" eingestellt. Nachdem ein neues Menü mit dem Dropdown-Menü Bearbeiten/Löschen erstellt wurde, kann das Standardenü aus der Bedienerschnittstellendatei gelöscht und ein neues Startmenü ausgewählt werden.

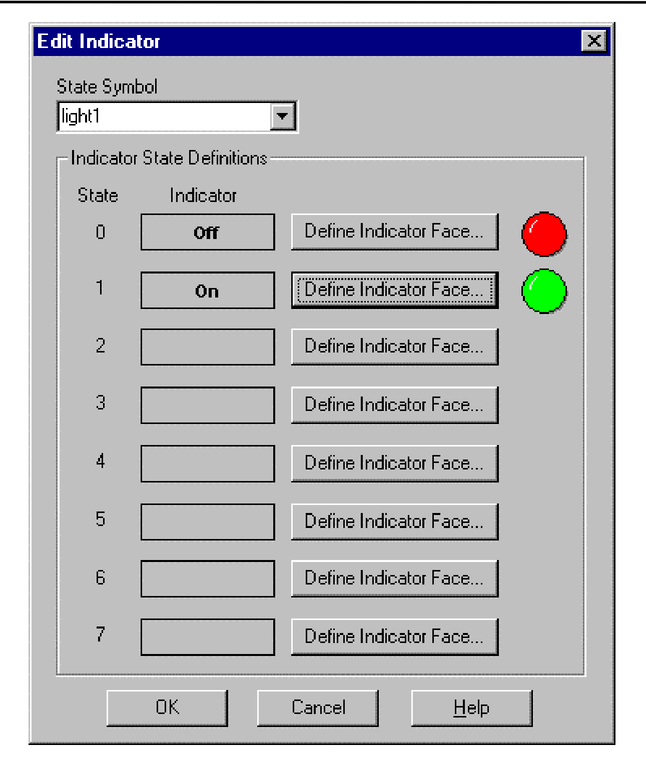
Bedienelemente hinzufügen

Kennmelder

Um den Status Ihrer Plätzchenbackmaschine anzuzeigen, können Sie noch einige Kennmelder hinzufügen. Erstellen Sie sechs Kennmelder, um den aktuellen Zustand der Lampen anzuzeigen, und einen, der Ihnen zeigt, wo Sie sich im Programm gerade befinden.



- Wählen Sie das Kennmelder-Werkzeug aus der Bedienerschnittstellen-Symbolleiste. Der Cursor wird zum Kennmelder-Cursor. Setzen Sie den Cursor auf den gewünschten Punkt im Bedienerschnittstellenmenü und klicken Sie die linke Maustaste, um einen Kennmelder abzulegen.
- Doppelklicken Sie auf den neuen Kennmelder, um die Dialogbox **Kennmelder bearbeiten** zu öffnen.
- Geben Sie im Listenfeld **Zustandssymbol** das Symbol "**light1**" ein oder wählen Sie es aus der Dropdown-Liste aus. Das Erscheinungsbild des Kennmelders ändert sich dann entsprechend dem aktuellen Wert von "light1".
- Klicken Sie auf die Schaltfläche **Kennmelder-Aussehen definieren** für Zustand 0, um die Dialogbox "Kennmelder definieren" zu öffnen.
- Geben Sie in der Editierbox"**Text:**" ein: "**AUS**". Klicken Sie dann auf die Schaltfläche "**Leuchtenfarbe wählen**" und wählen Sie ein rotes Symbol "Lampe".
- Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.
- Wiederholen Sie den gleichen Vorgang für Zustand 1. Geben Sie hier aber als Text "**EIN**" ein und wählen eine grüne Lampe. Am Ende sollte die Dialogbox wie folgt aussehen:



- Erstellen Sie mit **Kopieren/Einfügen** insgesamt sechs Kennmelder. Weisen Sie das **Zustandssymbol** den fünf neuen Kennmeldern **light2-light6** zu.
- Legen Sie über die Symbolleiste einen weiteren Kennmelder ab. Weisen Sie das **Zustandssymbol** dem neuen Kennmelder **State** zu.
- Weisen Sie den folgenden Text für die Zustände 0 bis 3 zu:
  Zustand 0 **Trog füllen**
  Zustand 1 **Mischen**
  Zustand 2 **Backen**
  Zustand 3 **Fertig**

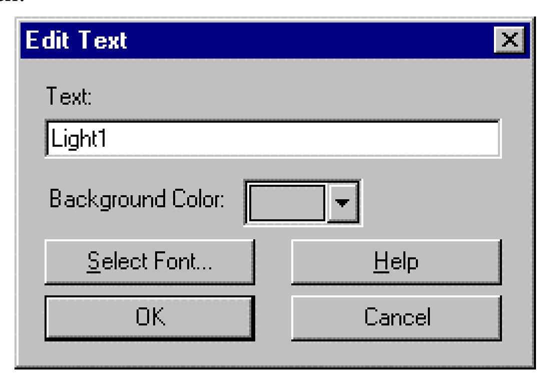
Textanzeige

Damit die Kennmelder nicht verwechselt werden, sollten Sie über jedem Kennmelder ein Textfeld anbringen



- Wählen Sie das Textwerkzeug aus der Bedienerschnittstellen-Symbolleiste. Der Cursor wird zum Text-Cursor. Setzen Sie den Cursor auf den Kennmelder für light1 und klicken Sie auf die linke Maustaste, um ein Textfeld abzulegen.
- Doppelklicken Sie auf das neue Textfeld, um die Dialogbox **Text bearbeiten** zu öffnen.
- Geben Sie in der Editierbox"**Text:**" ein: **Light1**.

- Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.
- Erstellen Sie fünf weitere Textfelder, um die Kennmelder für light2 bis light6 zu beschriften.



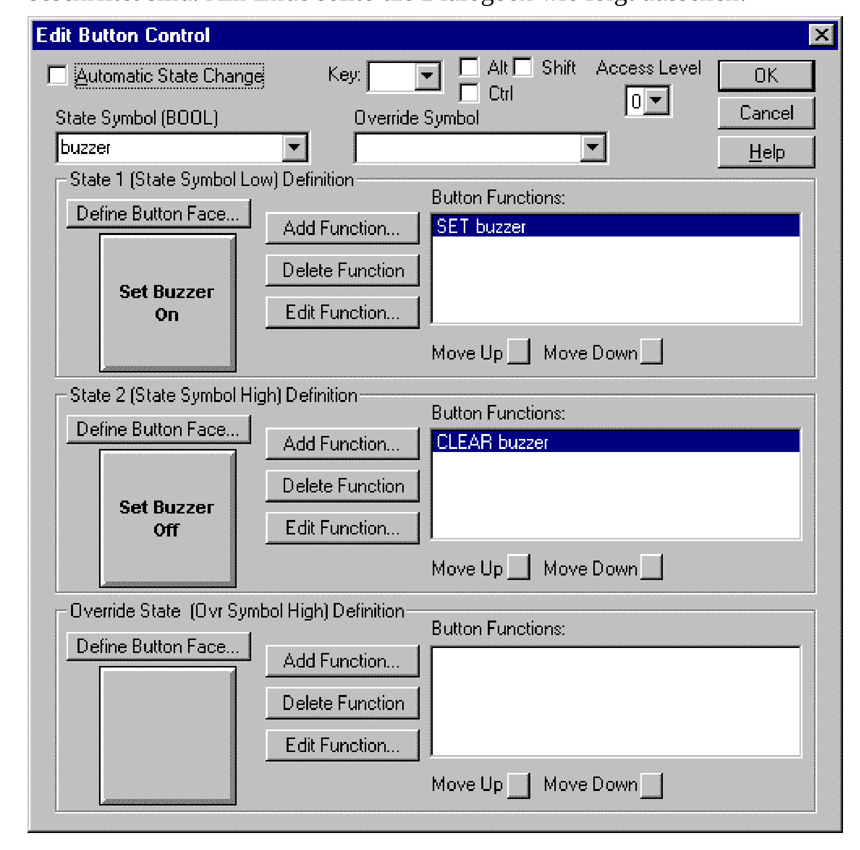
Klicktaste

Um den Summer "Fertig" in Ihrer Backmaschine steuern zu können, benötigen Sie noch eine Klicktaste.



- Wählen Sie das Klicktasten-Werkzeug aus der Bedienerschnittstellen-Symbolleiste. Der Cursor wird zum Schaltflächen-Cursor. Setzen Sie den Cursor auf den gewünschten Punkt im Bedienerschnittstellenmenü und klicken Sie die linke Maustaste, um eine Schaltfläche abzulegen.
- Doppelklicken Sie auf die neue Schaltfläche, um die Dialogbox **Schaltfläche bearbeiten** zu öffnen.
- Geben Sie in der Editierbox **Zustandssymbol** das Symbol **buzzer** ein.
- Klicken Sie im Teil **Definition Zustand 1** der Dialogbox auf die Schaltfläche **Funktion hinzufügen**.
- Wählen Sie aus dem aufgeblendeten Drop-Down-Listenfeld die Funktion **Einstellen**.
- Wählen Sie im nächsten Drop-Down-Listenfeld das Symbol **buzzer**.
- Klicken Sie im Teil **Definition Zustand 2** der Dialogbox auf die Schaltfläche **Funktion hinzufügen**.
- Wählen Sie aus dem aufgeblendeten Drop-Down-Listenfeld die Funktion **Löschen**.
- Wählen Sie im nächsten Drop-Down-Listenfeld das Symbol **buzzer**.

- Bearbeiten Sie mit den Tasten**Schaltfläche definieren** für die Zustände 1 und 2 die Schaltflächen so, daß sie mit "**Summer EIN**" bzw. "**Summer AUS**" beschriftet sind. Am Ende sollte die Dialogbox wie folgt aussehen:



- Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.

Haben Sie keinen E/A-Simulator, dann müssen Sie zwei weitere Klicktasten zur Steuerung der Symbole **switch1 und switch2** erstellen.

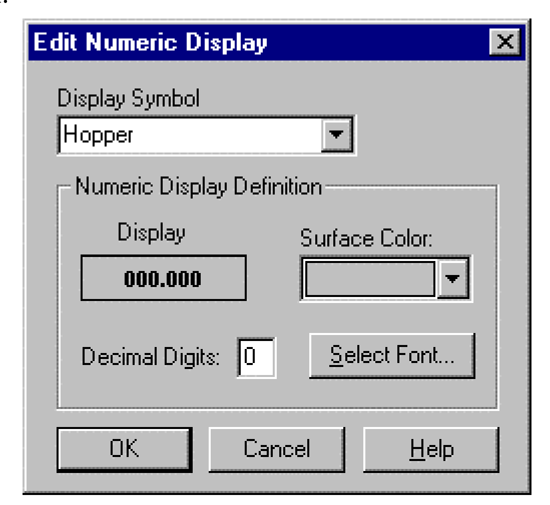
Zahlenanzeige

Fügen Sie eine Zahlenanzeige hinzu, über die Sie da Gewicht Ihres Einfülltrogs kontrollieren können.



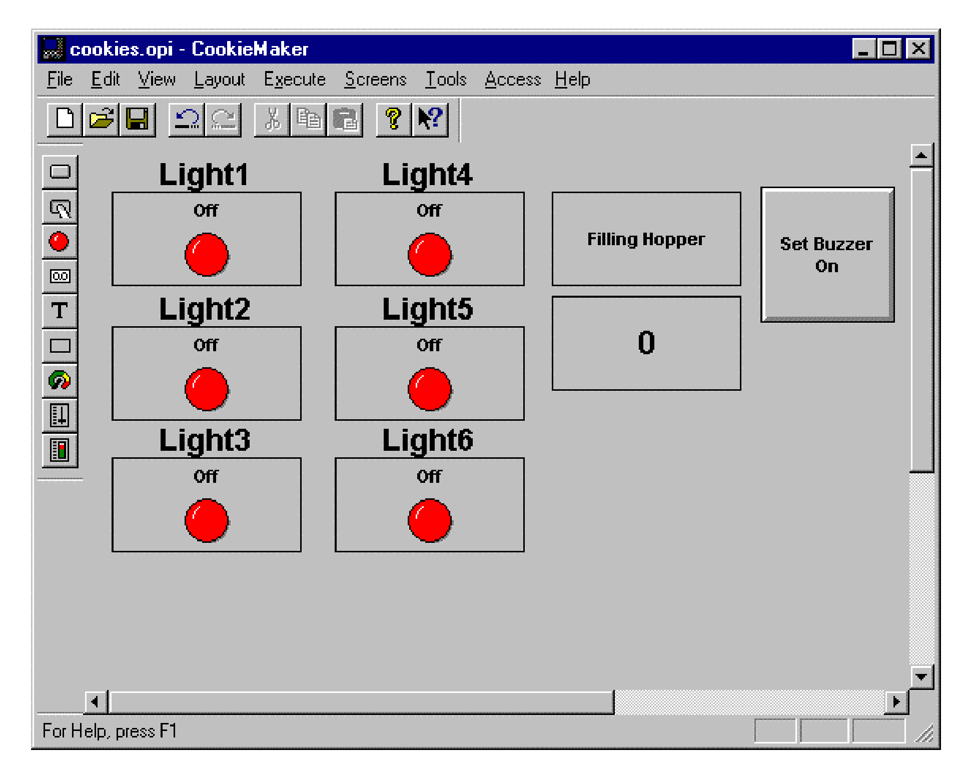
- Wählen Sie das Zahlenanzeigen-Werkzeug aus der Bedienerschnittstellen-Symbolleiste. Der Cursor wird zum Zahlen-Cursor. Setzen Sie den Cursor auf den gewünschten Punkt im Bedienerschnittstellenmenü und klicken Sie die linke Maustaste, um eine Zahlenanzeige abzulegen.
- Doppelklicken Sie auf die neue Zahlenanzeige, um die Dialogbox **Zahlenanzeige bearbeiten** zu öffnen.

- Geben Sie in der Editierbox **Anzeigesymbol** das Symbol **hopper** ein.
- Ändern Sie **Dezimalstellen** auf **0**.
- Klicken Sie auf **OK**, um die Dialogbox zu schließen und die Änderungen zu übernehmen.



Bedienermenüs aktivieren

Am Ende sollte Ihr Bedienermenü wie folgt aussehen:



Um das Menü im Betrieb zu zeigen, muß das Programm "**Cookie_maker.SFC**" laufen, wenn Sie den Menübefehl **Ausführen\Menüs aktivieren** wählen.

Anhang E Motion Control

Die Programmiersprache Motion Control ist ein Satz Textanweisungen für Bewegungssteuerungsoperationen, der RS-274D entspricht. Mit dieser Sprache können Sie zwei- oder dreidimensionale Bewegungen über Parameter steuern, die wie folgt aussehen:

- Koordinatenpositionen
- Vorschubgeschwindigkeiten
- Bewegungen zwischen Positionen mit gesteuerter Beschleunigung und Verzögerung

Die Programmierung in Motion Control besteht aus einer Reihe von Befehlen aus jeweils einem Buchstaben, dem numerische Parameter zu diesen Befehlen folgen. Die Befehle sind in einzelne Textzeilen aufgeteilt, den Sätzen. Diese Sätze bilden Bearbeitungseinheiten, die nacheinander ausgeführt werden. Die Programmausführung pausiert bei jedem Satz solange, bis alle Funktionen in dem Satz ausgeführt sind. Dann wird der Programmfluß mit dem nächsten Satz fortgesetzt.

Sie können einer Ablaufkette Elemente in Motion Control hinzufügen und strukturierten Text in einen Code aus Motion Control einbetten. Weitere Informationen hierzu finden Sie unter "Motion Control Anweisungen” auf Seite E-4.

E

Motion Control konfigurieren

PC Control kann mit den nachstehenden Schnittstellenmodulen verwendet werden, die Motion Control unterstützen:

Delta Tau PMAC-PC

Delta Tau PMAC Direct

Delta Tau PMAC2

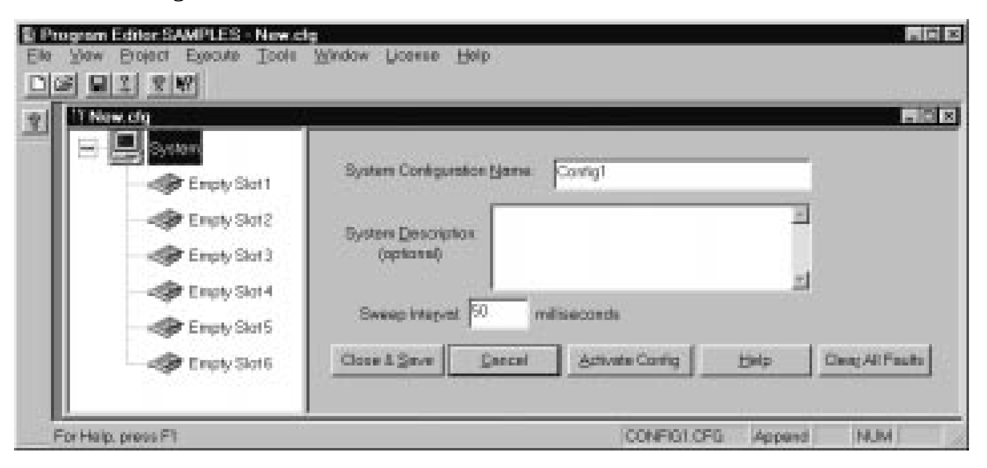
Motion Engineering PCDSP

Motion Options Compumotor

Hinweis

In einer Systemkonfiguration ist jeweils nur *ein* Bewegungsmodul erlaubt.

A. Klicken Sie im PC Control Programmeditor-Dateimenü auf "Konfiguration öffnen" (oder "Neue Konfiguration"). Hierauf erscheint folgendes Konfigurationsfenster:



B. Wählen Sie aus der Baumanzeige auf der linken Seite des Fensters den Steckplatz aus, den Sie konfigurieren wollen.

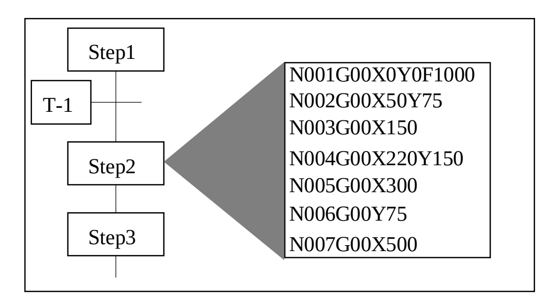
C. Wählen Sie aus dem Feld "Modultyp" das bei Ihnen installierte Bewegungsmodul aus. Die Konfigurations-Dialogbox erscheint für das ausgewählte Modul.

D. Klicken Sie auf die Schaltfläche "Moduldefinition". Einzelheiten zum Erstellen der Konfiguration finden Sie in der Online-Hilfe zu Ihrem Motion Control Schnittstellenmodul.

Motion Control Programmierung

Einer Ablaufkette Motion Control hinzufügen

Bei der Erstellung des Anwendercodes zu einem Ablaufketten-Schritt können Sie wahlweise Code in Motion Control verwenden (siehe nachstehende Abbildung). Wird die Ablaufkette bearbeitet, dann wird der in den einzelnen Schritten eingegliederte Motion Control Code verarbeitet, sobald der Schritt aktiv wird. Sie können den Motion Control Code entweder direkt in einen Schritt eingeben oder bei der Konfiguration des Schrittes eine Datei, die den Motion Control Code enthält, mit dem Schritt verknüpfen. Der Datentyp in er Datei muß dabei vom gleichen Format sein wie der Typ der direkt in den Schritt eingegebenen Daten.



Wie die Steuerungssystem-Software die RS-274D Spezifikation erweitert wird in "PC CONTROL Software Erweiterungen gegenüber RS-274D" auf Seite E-3 erläutert.

PC CONTROL Software Erweiterungen gegenüber RS-274D

Die Steuerungssystem-Software erweitert die RS-274D Spezifikation um folgende Punkte:

- Sie können im Motion Control Code Zuweisungen und Ausdrücke in strukturiertem Text einbetten. Siehe "Strukturierten Text in Motion Control Code einbetten" auf Seite E-27.
- Sie können Schrittaktionen einsetzen, um die E/A-Operationen mit der Ausführung von Motion Control zu synchronisieren. Über Bewegungs-Bestimmungszeichen können Sie Aktionslogik mit Bewegung synchronisieren.
- Die Steuerungssystem-Software stellt Bewegungssymbole zur Verfügung, die auf der E/A abgebildet werden. So ist zum Beispiel "axis.CMDPOS" ein Symbol vom Typ reelle Zahl, das die aktuell befohlene Position einer Achse enthält.
- M-Merker sind mit den Motion Control M-Codes verknüpft und werden entsprechend dem eingesetzten M-Code auf EIN oder AUS gesetzt. Hierdurch

können Sie die Merker von anderen Programmen aus überwachen und andere Aktionen starten, nachdem bei der Programmbearbeitung ein M-Code angetroffen wurde.

- Mit den Befehlen WHILE und IF-GOTO können Sie die Steuerung des Programmablaufs bei den Motion Control Befehlen verbessern.
- Mit einem Wegbefehl G65 können Sie ein Makro (Unterprogramm) zur Ausführung aufrufen.

Motion Control Anweisungen

Verwendung von Motion Control Befehlen

Die Steuerungssystem-Software wandelt RS-274D Befehle in die entsprechenden Befehle um, die speziell für das von Ihnen ausgewählte Servomodul erzeugt werden.

Fahranweisungen müssen in einem speziellen Format strukturiert werden. Siehe "Motion Control Satzformat” auf Seite E-5. Die Fahranweisungen sind:

Anweisung	Funktion
A	Rotation X Fahranweisung in vordefinierten technischen Einheiten.
B	Rotation Y Fahranweisung in vordefinierten technischen Einheiten.
C	Rotation Z Fahranweisung in vordefinierten technischen Einheiten.
D	Werkzeugfunktion zur Auswahl der Werkzeugkompensation.
F	Vorschubfunktion oder Geschwindigkeitsbefehl in vordefinierten technischen Einheiten.
G	Wegfunktion. Siehe "Verwendung von G-Codes” auf Seite E-6.
I	Interpolationsparameter oder Gewindesteigung parallel zur X-Achse.
J	Interpolationsparameter oder Gewindesteigung parallel zur Y-Achse.
K	Interpolationsparameter oder Gewindesteigung parallel zur Z-Achse.
M	Zusatzfunktion. Siehe "Verwendung von M-Codes” auf Seite E-8.
N	Satznummer (wahlweise)
S	Spindeldrehzahlfunktion.
T	Werkzeugfunktion. Bereich 01-32.
X	Lineare X Fahranweisung in vordefinierten technischen Einheiten.
Y	Lineare Y Fahranweisung in vordefinierten technischen Einheiten.
Z	Lineare Z Fahranweisung in vordefinierten technischen Einheiten.

Motion Control Satzformat

Das Format eines Motion Control Satzes (einzelne Codezeile) wird nachstehend beschrieben.

N Falls eine Sequenznummer verwendet wird, muß sie am Anfang des Satzes stehen. Wahlfrei für das Steuerungssystem.

G Die Wegfunktionen G müssen auf N folgen.

X Die linearen Dimensionsworte folgen auf G. Geben Sie zuerst die X-Achse an.

Y Die linearen Dimensionsworte folgen auf G. Geben Sie die Y-Achse als zweites an.

Z Die linearen Dimensionsworte folgen auf G. Geben Sie die Z-Achse als drittes an.

A Die rotatorischen Dimensionsworte folgen auf G. Geben Sie zuerst die X-Achse an.

B Die rotatorischen Dimensionsworte folgen auf G. Geben Sie die Y-Achse als zweites an.

C Die rotatorischen Dimensionsworte folgen auf G. Geben Sie die Z-Achse als drittes an.

I Die Interpolationsworte folgen den Dimensionsworten. Geben Sie zuerst die X-Achse an.

J Die Interpolationsworte folgen den Dimensionsworten. Geben Sie die Y-Achse als zweites an.

K Die Interpolationsworte folgen den Dimensionsworten. Geben Sie die Z-Achse als drittes an.

D Die Auswahl der Werkzeugkompensation muß auf K folgen.

F Geben Sie eine Vorschubgeschwindigkeit ein, die für mehrere Achsen gilt, muß F auf das letzte Dimensionswort (und Interpolation) folgen, auf das es zutrifft..

T Die Auswahl der Werkzeugfunktion folgt auf S.

M Alle Zusatzfunktionen müssen im letzten Satz unmittelbar vor dem Satzendezeichen stehen.

Satzende Zeigt das Ende eines Satzes mit dem Zeichen "Wagenrücklauf / Zeilenvorschub" an. Zwischen dem letzten Befehl und dem Blockende dürfen keine weiteren Leerzeichen, Tabulatoren oder sonstige Zeichen stehen.

Beispiele von Motion Control Sätzen

Beispiel 1

N009G01X-3.0Y-7.0Z+1.0F95

N009 Neunter Satz im Programm.

G01 Positioniert ein Werkzeug auf den nächsten Punkt entlang einer Geraden.

Stellt die Position der X-Achse ein.

Stellt die Position der Y-Achse ein.

Stellt die Position der Z-Achse ein.

F95 Stellt die Vorschubgeschwindigkeit auf 95 Einheiten ein.

Beispiel 2

N011G02X+0.5Y+1.0I0.75J0.0

N011 Elfter Satz im Programm.

G02 Positioniert ein Werkzeug in einer Kreisbewegung.

Stellt die Position der X-Achse ein.

Stellt die Position der Y-Achse ein.

I0.75 Stellt die Position der X-Achse für den Kreisbogen ein.

J0.0 Stellt die Position der Y-Achse für den Kreisbogen ein.

Beispiel 3

N003G70X+1.3Y-7.0Z+2.1M08

N003 Dritter Satz im Programm.

G70 Stellt die Betriebsart für Programmierung im Zollmaß ein.

Stellt die Position der X-Achse ein.

Stellt die Position der Y-Achse ein.

Stellt die Position der Z-Achse ein.

M08 Schaltet Kühlmittel Nummer 1 an.

Verwendung von G-Codes

Die Steuerungssystem-Software unterstützt RS-274D G-Codes zur Steuerung zahlreicher Bewegungssteuerungsoperationen. Die Unterstützung des G-Codes ist treiberabhängig. Nicht alle G-Codes werden von allen Treibern unterstützt. In der nachstehenden Tabelle zeigt eine Markierung (•) an, welche G-Codes von welchen Treibern unterstützt werden.

Die verfügbaren G-Codes sind in der nachstehenden Tabelle aufgeführt:

Code	Beschreibung	Bemerkung	PMAC	PCDSP	Compu-motor
G00	Schnelle Punkt-zu-Punkt-Bewegung	Die Achsen kommen nicht unbedingt zusammen am Endpunkt an.	•	•	•
G01	Koordinierte Linearbewegung	Die Bewegungsgeschwindigkeit wird geregelt und die Bewegung aller Achsen ist koordiniert.	•	•	
G02	Kreisbewegung im Uhrzeigersinn	Die Achsen bewegen sich koordiniert im Uhrzeigersinn (von oben gesehen).	•	Nur die ersten beiden Achsen	
G03	Kreisbewegung, im Gegenuhrzeigersinn	Die Achsen bewegen sich koordiniert im Gegenuhrzeigersinn (von oben gesehen).	•	Nur die ersten beiden Achsen	
G04	Verweilzeit	Die Programmausführung wird für die programmierte Zeitdauer ausgesetzt. Die Dauer wird durch ein F-Wort im gleichen Satz angegeben.. (z.B. G04F2.5 = Verzögerung von 2,5 Sekunden).	•	•	•
G17	Xp - Yp Ebenenwahl,	mit Xp: X-Achse oder eine Parallelachse.	•		
G18	Zp - Xp Ebenenwahl	mit Yp: Y-Achse oder eine Parallelachse.	•		
G19	Yp - Zp Ebenenwahl	mit Zp: Z-Achse oder eine Parallelachse.	•		
G40	Abbruch Radiuskompensation		•		
G41	Werkzeugdurchmesserkompensation (links)	Dreidimensionale Kompensation.	•		
G42	Werkzeugdurchmesserkompensation (rechts)	Dreidimensionale Kompensation.	•		
G43	Werkzeuglängen-Offset (plus)		•		
G44	Werkzeuglängen-Offset (minus)		•		
G45	Werkzeugoffset erhöhen		•		
G46	Werkzeugoffset verringern		•		
G47	Werkzeugoffset verdoppeln		•		
G48	Werkzeugoffset halbieren		•		
G52	Lokaler Offsetkoordinaten-Nullpunkt	Anwendungsspezifisch	•		
G53	Bewegung im Maschinenkoordinatensystem		•	•	
G54	Werkstückkoordinatensystem 1		•		
G55	Werkstückkoordinatensystem 2		•		
G56	Werkstückkoordinatensystem 3		•		
G57	Werkstückkoordinatensystem 4		•		

Code	Beschreibung	Bemerkung	PMAC	PCDSP	Compu-motor
G58	Werkstückkoordinatensystem 5		•		
G59	Werkstückkoordinatensystem 6		•		
G61	Genauhalt-Modus		•		
G64	Schnittmodus		•		
G65	Makroaufruf		•	•	•
G66	Durchgangsfunktion	Treiberspezifisch	•	•	
G70	Zollmodus	Definiert während der Konfiguration, nicht verfügbar während der Laufzeit.	Nicht in Laufzeit	Nicht in Laufzeit	
G71	Metrischer Modus	Definiert während der Konfiguration, nicht verfügbar während der Laufzeit.	Nicht in Laufzeit	Nicht in Laufzeit	
G90	Absolute Positionierung		•	•	
G91	Kettenmaßpositionierung		•	•	
G92	Position voreingestellt	Die Achsenpositionen werden auf die im Satz angegebenen Werte eingestellt.	•	•	
G93	Zeitreziproker Vorschub		•		
G94	Vorschub pro Minute		•	Nur Vorschub pro Minute	

Verwendung von M-Codes

M-Codes sind anwenderdefinierte Operationen, die in RS-274D und AS unterstützt werden. Zulässige M-Codes sind M0 bis M99. Die M-Codes laufen in RS-274D und signalunterstützender Logik in auszuführenden AS- oder KOP-Programmen. Für jeden M-Code existieren die entsprechenden Steuerungssystemsymbole Mflag0 bis Mflag98. Geradzahlig numerierte M-Codes schalten die M-Merker aus, während ungerade numerierte Codes den entsprechenden Merker Einschalten. Zum Beispiel:

RS-274D Anweisung	Ergebnis
M10	Mflag10 FALSE
M11	Mflag10 TRUE
M96	Mflag96 FALSE
M97	Mflag97 TRUE

Hinweise:

Spezial-M-Codes müssen allein in einer Zeile stehen. Hierzu gehören M00, M01, M03, M04, M05, M58 und M59.

M-Codes, die in der gleichen Zeile wie eine Bewegung stehen, werden vor der Bewegung ausgeführt.

Vordefnierte M-Codes

Die Steuerungssystem-Software führt für einige M-Codes eine Vordefinition für interne Operationen durch. Diese können dann nicht zum Ein- oder Ausschalten von M-Merkern verwendet werden:

Code:	Beschreibung:
M0	Programmierter Halt
M1	Wahlweiser Programmhalt
M2	Programmende
M3	Spindel im Uhrzeigersinn
M4	Spindel im Gegenuhrzeigersinn
M5	Spindel-Halt
M30	Programmende mit Rücksetzen
M99	Ende Makrofunktion

Warten und fortsetzen - M-Code

Die Steuerungssystem-Software kann auf RS-274D M-Codes warten. Nachdem der M-Code bearbeitet ist, läßt die Steuerungssystem-Software das Anwenderprogramm solange warten, bis Ihre Logik das System darüber informiert hat, daß die Operation beendet ist und die Ausführung mit dem nächstfolgenden RS-274D Satz. Um diese Funktion freizugeben müssen Sie das Kästchen "Auf alle M-Codes warten" auf der Bewegungsoptionen-Konfigurationsseite aktivieren..

Diese Steuerungssystemsymbole betreffen die M-Codes "Warten und Fortsetzen":

Variablenname	Typ	Verwendung	Lesen (R) & Schreiben (W)	gelöscht (C) / gesetzt (S) von	
				Steuerungs-S/W	Anwender
AxisGroup.MWAIT	BOOL	Warten auf M-Code-Bearbeitung	R	S/C	____
AxisGroup.MCODE	INT	Wert von M-Code	R	S/C	____
AxisGroup.MCONT	BOOL	M-Code-Bearbeitung beendet – Ausführung wieder aufnehmen	R/W	C	S

Verwendung der Funktion "M-Merkersymbole definieren"

Wenn Sie mit der Steuerungssystem-Software 46 globale Symbole des Typs BOOL erzeugen wollen, dann wählen Sie die Auswahlbox "M-Merkersymbole definieren" Die Symbole werden erzeugt, wenn diese Option ausgewählt und die Konfiguration gespeichert und aktiviert ist. Diese Symbole erscheinen im Symbolmanager, sie besitzen die Symbolnamen Mflag6 bis Mflag96 mit gerader Numerierung.

Verwendung der Funktion "Auf alle M-Codes warten"

Wählen Sie die Auswahlbox "Auf alle M-Codes warten", wenn Ihr Anwenderprogramm die Ausführung von RS-274D solange unterbrechen soll, bis Ihre M-Codelogik anzeigt, daß die Aktion beendet ist. Anschließend kann der Rest des Programms ausgeführt werden. Wird ein M-Code ausgeführt, dann wird das Bewegungsprogramm solange angehalten und der Merker axis_group.MWAIT aktiviert, bis das Anwenderprogramm den Merker axis_group.MCONT setzt. Hierdurch wird das Bewegungsprogramm fortgesetzt und die Merker axis_group.MWAIT und axis_group.MCONT werden von der Steuerung rückgesetzt. Das Anwenderprogramm ist dafür verantwortlich, daß Logik zur Verfügung steht, mit der der Merker axis_group.MCONT gesetzt wird, nachdem der Merker axis_group.MWAIT aktiviert wurde.

axis_group.MCODE ist ein ganzzahliger Wert, der den Wert des M-Code enthält. axis_group ist der Name der Achsgruppe, mit der der M-Code verknüpft ist. Definieren Sie die Achsgruppenbezeichnungen in der Projektkonfigurationsdatei.

Verwendung der Funktion "M-Codes nicht bearbeiten"

Die Steuerungssystem-Software unterstützt die spezielle M-Codefunktion entsprechend der Beschreibung in der RS-274D Spezifikation. Wenn Sie auf dieses Häkchen klicken ignoriert das Anwenderprogramm die folgenden M-Codes:

- M3 (Spindel im Uhrzeigersinn)
- M4 (Spindel im Gegenuhrzeigersinn)
- M5 (Spindel-Halt)

Verwendung vordefinierter Motion Control Symbole

Die Steuerungssystem-Software liefert vordefinierte Symbole, mit denen Sie einige Ihrer Bedien- und Beobachtungsfunktionen ausführen können. . Verknüpfen Sie bei der Konfiguration der E/A die Symbole mit den entsprechenden Funktionen.

Die vordefinierten Symbole sind wie folgt gruppiert:

- Achsenausgangssymbole
- "Achsgruppenausgangssymbole” auf Seite E-12
- "Spindelausgangssymbole” auf Seite E-13
- "Achseneingangssymbole” auf Seite E-13
- "Achsgruppeneingangssymbole” auf Seite E-16

Hinweis: Nicht alle vordefinierten Achssymbole werden von allen Bewegungstreibern unterstützt. Schauen Sie in der Hilfe zu den Bewegungstreibern nach unterstützten Achssymbolen.

Achsenausgangssymbole

Symbol	Datentyp	Funktion
axis.ACTPOS	REAL	Enthält die Istposition der Achse.
axis.ACTVEL	REAL	Enthält die Istgeschwindigkeit der Achse.
axis.CMDPOS	REAL	Enthält die Sollposition der Achse.
axis.TPOS	REAL	Enthält die aktuelle Zielposition der Achse.
axis.AXSFE	REAL	Enthält den Nachlauffehler der Achse.
axis.A	BOOL	Zeigt an, ob die Achse freigegeben ist.
axis.IP	BOOL	Zeigt an, ob die Achse in Position ist.
axis.MC	BOOL	Zeigt an, ob die Achsbewegung abgeschlossen ist.
axis.ESTPO	BOOL	Zeigt an, ob der Achsen-NOTHALT aktiviert ist.
axis.RELPOS	REAL	Enthält die Relativposition der Achse.
axis.HOME	BOOL	Zeigt an, daß die Achse in Grundstellung ist.
axis.STATUS	REAL	Zeigt Achszustand an - Bitdefinitionen sind treiberabhängig. Siehe Treiberhilfe.
axis.CMDSPD	REAL	Enthält die Sollgeschwindigkeit der Achse.

Achsgruppenausgangssymbole

Symbol	Datentyp	Funktion
AxisGroup.ESTPO	BOOL	Zeigt, ob Nothalt aktiviert wurde.
AxisGroup.TOOL	Integer	Enthält das aktive Werkzeugoffset.
AxisGroup.INTPL	Integer	Enthält aktiven Interpolationsmodus für Gruppe. 0=Punkt-zu-Punkt 1=linear 2=zirkular im Uhrzeigersinn 3=zirkular im Gegenuhrzeigersinn.
AxisGroup.CIR	BOOL	Zeigt, ob Zirkularinterpolation (im Uhrzeigersinn oder im Gegenuhrzeigersinn) aktiviert wurde.
AxisGroup.PTP	BOOL	Zeigt, ob Punkt-zu-Punkt-Interpolation aktiviert wurde.
AxisGroup.LIN	BOOL	Zeigt, ob Linearinterpolation aktiviert wurde.
AxisGroup.CIRCW	BOOL	Zeigt, ob Zirkularinterpolation im Uhrzeigersinn aktiviert wurde.
AxisGroup.CIRCCW	BOOL	Zeigt, ob Zirkularinterpolation im Gegenuhrzeigersinn aktiviert wurde.
AxisGroup.DWL	BOOL	Zeigt, ob Verweilzeit aktiviert wurde.
AxisGroup.DWLTIM	Integer	Enthält die Verweilzeit, in ms heruntergezählt.
AxisGroup.ENG	BOOL	Zeigt, ob Zollmaße aktiviert sind.
AxisGroup.METRIC	BOOL	Zeigt, ob metrische Einheiten aktiviert sind.
AxisGroup.ABSDIM	BOOL	Zeigt, ob Bezugsmaßprogrammierung aktiviert ist.
AxisGroup.INCDIM	BOOL	Zeigt, ob Kettenmaßprogrammierung aktiviert ist.
AxisGroup.WAIT	BOOL	Zeigt, daß das Programm zum Werkzeugwechsel angehalten wurde.
AxisGroup.CONT	BOOL	Zeigt, daß Programm nach Werkzeugwechsel fortgesetzt wurde.
AxisGroup.MWAIT	BOOL	Zeigt, daß auf M-Code-Bearbeitung gewartet wird.
AxisGroup.MCODE	Integer	Enthält den Wert des M-Codes.
AxisGroup.PLANE	Integer	Aktive Ebene für Zirkularinterpolation. 0=FIXOFF[0] (G54) 1=FIXOFF[1] (G55) 2=FIXOFF[2] (G56) 3=FIXOFF[3] (G57) 4=FIXOFF[40] (G58) 5=FIXOFF[5] (G59)
AxisGroup.CUTMOD	Integer	Aktiver Bearbeitungsmodus. 0=Schnittmodus (G64). 1=Genauhalt (G61).
AxisGroup.FEDMOD	Integer	Aktiver Vorschubgeschwindigkeitsmodus. 0=Vorschubgeschwindigkeitsprogrammierung IPM, DPM (G94). 1=Zeitreziproke Programmierung (G93).

Spindelausgangssymbole

Symbol	Datentyp	Funktion
S.WAIT	BOOL	Zeigt an, daß das Programm für einen Spindelwechsel angehalten wurde.
S.CONT	BOOL	Zeigt an, daß das Programm nach einem Spindelwechsel fortgesetzt wurde.
S.DIR	BOOL	Zeigt die programmierte Spindel-Drehrichtung an. 0=positiv (im Uhrzeigersinn/M3) 1=negativ (im Gegenuhrzeigersinn/M4)
S.CMDSPD	REAL	Enthält die programmierte Spindeldrehzahl. Durch letzten S-Befehl eingestellt.

Der Gruppenstatus für Motion Control Befehle wird zu der Gruppe geschrieben, die die angegebene Achse enthält. Wird keine Achse angegeben, wird der Achsgruppenstatus zu axisGroup#1, der Standard-Achsgruppe, geschrieben.

Achseneingangssymbole

Symbol	Datentyp	Funktion
axis.JP	BOOL	Veranlaßt Tippen in positiver Richtung. Siehe "Verwendung der Achseneingangssymbole .JM und .JP” auf Seite E-15.
axis.JM	BOOL	Veranlaßt Tippen in negativer Richtung. Siehe "Verwendung der Achseneingangssymbole .JM und .JP” auf Seite E-15.
axis.HLD	BOOL	Stoppt die Achsbewegung, bis das Bit rückgesetzt ist.
axis.STP	BOOL	Stoppt die Achsbewegung.
axis.JSPD	REAL	Stellt Tippgeschwindigkeit für Achse ein.
axis.JINCR	REAL	Stellt Schrittmaß für Achse ein.
axis.JTYPE	Integer	Stellt Tippart für Achse ein (0=Referenzpunkt, 1=kont., 2=inkr.).
axis.SOVR	REAL	Stellt Drehzahlkorrektur für Achse ein (100.0=100%).
axis.TOOLOFF[x]	REAL	Ein Feld von Werkzeugoffsets für Achse. x = 0-9. Siehe "Verwendung der Achseneingangssymbole .TOOLOFF” auf Seite E-14. (Nicht für MEI.)
axis.FIXOFF[x]	REAL	Ein Feld von Vorrichtungsoffsets für Achse. x = 0-5. Siehe auch, "Verwendung der Achseneingangssymbole .FIXOFF ” auf Seite E-14. (Nicht für MEI.)
axis.RAPID	REAL	G00 Drehzahl.

Verwendung der Achseneingangssymbole .FIXOFF

Mit den Wegbefehlen G-55 bis G-59 werden Vorrichtungsoffsets eingestellt. G-54 löscht Vorrichtungsoffsets. Die Achseneingangssymbole axis.FIXOFF enthalten die von diesem Befehlen verwendeten Offsetwerte. Das folgende Beispiel zeigt den Zusammenhang zwischen den Befehlen und den Symbolen, die die entsprechenden Offsets enthalten:

G -Befehl	Symbol
G-54	axis.FIXOFF[0]
G-55	axis.FIXOFF[1]
G-56	axis.FIXOFF[2]
G-57	axis.FIXOFF[3]
G-58	axis.FIXOFF[4]
G-59	axis.FIXOFF[5]

Weisen Sie den Feldelementen in einem strukturierten Textschritt, der vor dem Motion Control Schritt mit den G-Befehlen steht, Werte zu.

Verwendung der Achseneingangssymbole .TOOLOFF

Mit den Wegbefehlen G-45 bis G-48 werden Werkzeugoffsets eingestellt. Bei allen vier Befehlen muß der D-Befehl im gleichen Satz verwendet werden. Der D Befehl gibt an, welches axis.TOOLOFF[ ] Element gewählt werden muß. Die Achseneingangssymbole axis.TOOLOFF enthalten die von den D-Befehlen verwendeten Offsetistwerte. Das folgende Beispiel zeigt den Zusammenhang zwischen den D-Befehlen und den Symbolen, die die entsprechenden Offsetwerte enthalten:

D-Befehl	Symbol
D-0	axis.TOOLOFF[0]
D-1	axis.TOOLOFF[1]
.	.
D-8	axis.TOOLOFF[8]
D-9	axis.TOOLOFF[9]

Weisen Sie den Feldelementen in einem strukturierten Textschritt, der vor dem Motion Control Schritt mit den G- und D-Befehlen steht, Werkzeugoffsetwerte zu.

Verwendung der Achseneingangssymbole .JM und .JP

Die Achseneingangssymbole axis.JP und axis.JM sind Boolesche Symbole, die bewirken, daß sich die Achse im Tippbetrieb in positive bzw. negative Richtung bewegt. Für diese Symbole gelten folgende Bedingungen:

- Die Symbole axis.JP und axis.JM haben erst eine Funktion, nachdem Sie den Achseneingangssymbolen axis.JSPD und axis.JTYPE Werte zugewiesen haben, mit denen Tippgeschwindigkeit und Tippart spezifiziert werden.
- Die durch die Symbole axis.JP und axis.JM angesprochenen Achsen müssen in einer Gruppe sein, die keine weiteren Achsen enthält. Falls in der Gruppe doch noch weitere Achsen enthalten sind, können sie nicht von einem anderen Bewegungssteuerungsprogramm gesteuert werden, wenn ein Tippbefehl zu der von den Symbolen axis.JP und axis.JM angesprochenen Achse ausgegeben wird..

Referenzpunktfahren einer Achse

1. Stellen Sie ein .JTYPE = 0.
2. Stellen Sie ein .JSPD = Geschwindigkeit zum Referenzpunktfahren.
3. Aktivieren Sie entweder .JP oder .JM.
4. Warten Sie, bis .HOME = TRUE für mehr als 1 Sekunde.
5. Setzen Sie .JP oder .JM zurück.

Achsgruppeneingangssymbole

Symbol	Datentyp	Funktion
AxisGroup.ESTP	BOOL	Aktiviert den NOTHALT. Das Achseneingangssymbol axis.ESTP ist ein Boolescher Wert, der alle definierten Achsen anhält. Beachten Sie, daß dies alle Achsen in allen Gruppen betrifft, unabhängig davon, welche Achse von dem Symbol adressiert wird.
AxisGroup.HLD	BOOL	Stoppt die Gruppenbewegung, bis das Bit rückgesetzt ist.
AxisGroup.STP	BOOL	Stoppt die Gruppenbewegung.
AxisGroup.SOVR	REAL	Drehzahlkorrektur für Gruppe (100.0=100%).
AxisGroup.MCONT	BOOL	M-Code-Bearbeitung beendet.
AxisGroup.TOOLRAD[x]	REAL	Wert für Werkzeugradiuskompensation (G41 und G42). x = 0-31. Weitere Informationen finden Sie unter "Verwendung der Achsgruppeneingangssymbole .TOOLRAD ” auf Seite E-16
AxisGroup.TOOLLEN[x]	REAL	Wert für Werkzeuglängenkompensation (G43 und G44). x = 0-31. Weitere Informationen finden Sie unter "Verwendung der Achsgruppeneingangssymbole .TOOLLEN” auf Seite E-17.
AxisGroup.RESTP	BOOL	Setzt NOTHALT zurück, bis das Bit rückgesetzt ist. HINWEIS: Dieser Eingang sollte nur gepulst werden. Bleibt er ständig an, reagiert das System nicht mehr auf andere Eingangssignale.
AxisGroup.OPSTP	BOOL	Aktiviert den wahlweisen Halt (M01).
AxisGroup.TSTRUN	BOOL	Aktiviert Testlaufmodus.

Verwendung der Achsgruppeneingangssymbole .TOOLRAD

Über die Wegbefehle G-41 und G-42 werden Schneiden- oder Radiuskompensation auf den Werkzeugpfad angewandt. Bei beiden Befehlen muß der D-Befehl im gleichen Satz verwendet werden. Der D Befehl gibt an, welches axisGroup.TOOLRAD[ ] Element gewählt werden muß. Die Achsgruppeneingangssymbole axisGroup.TOOLRAD enthalten die von den D-Befehlen verwendeten Kompensationsistwerte. Das folgende Beispiel zeigt den Zusammenhang zwischen den D-Befehlen und den Symbolen, die die entsprechenden Kompensationswerte enthalten: Wegbefehle sind G-Codes. Weitere Informationen hierzu finden Sie unter "Verwendung von G-Codes” auf Seite E-6.

Zusammenhang zwischen D-Befehlen und Symbolen

D-Befehl	Symbol
D-0	axisGroup.TOOLRAD[0]
D-1	axisGroup.TOOLRAD[1]
D-2	axisGroup.TOOLRAD[2]
.	.
.	.
.	.
D-29	axisGroup.TOOLRAD[29]
D-30	axisGroup.TOOLRAD[30]
D-31	axisGroup.TOOLRAD[31]

Weisen Sie den Feldelementen in einem strukturierten Textschritt, der vor dem Motion Control Schritt mit den G- und D-Befehlen steht, Kompensationswerte zu.

Der Code G40 hebt die Kompensation auf.

Verwendung der Achsgruppeneingangssymbole .TOOLLEN

Über die Wegbefehle G_-43H und G_-44H werden Werkzeuglängenkompensation auf den Werkzeugpfad angewandt. H gibt an, welches axisGroup.TOOLLEN Element gewählt werden muß. Die Achsgruppeneingangssymbole axisGroup.TOOLLEN enthalten die von H verwendeten Kompensationsistwerte. Da nachstehende Beispiel zweigt den Zusammenhang zwischen H und den Symbolen, die ihre entsprechenden Kompensationswerte enthalten. Wegbefehle sind G-Codes. Weitere Informationen hierzu finden Sie unter "Verwendung von G-Codes” auf Seite E-6.

D-Befehl	Symbol
H0	axisGroup.TOOLLEN[0]
H1	axisGroup.TOOLLEN[1]
H2	axisGroup.TOOLLEN[2]
.	.
.	.
.	.
H29	axisGroup.TOOLLEN[29]
H30	axisGroup.TOOLLEN[30]
H31	axisGroup.TOOLLEN[31]

Weisen Sie den Feldelementen in einem strukturierten Textschritt, der vor dem Motion Control Schritt mit den G-Befehlen steht, Kompensationswerte zu.

G40 hebt die Kompensation auf.

Konfiguration von Bewegungsoptionen

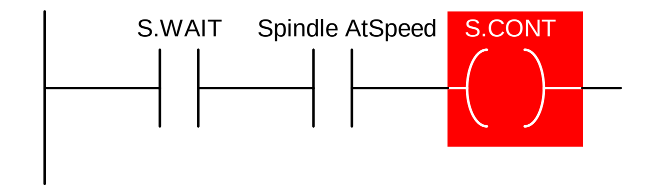
Die Steuerungssystem-Software enthält die folgenden Funktionen, die Sie bei der Konfiguration der Bewegungsoptionen unterstützen: c

- "Verwendung der Funktion "Anhalten bei Spindelbefehlen"" auf Seite E-18
- "Verwendung der Funktion "Anhalten bei Werkzeugwechsel"" auf Seite E-19
- "Verwendung der Funktion "M-Merkersymbole definieren"" auf Seite E-10
- "Verwendung der Funktion "Auf alle M-Codes warten"" auf Seite E-10
- "Verwendung der Funktion "M-Codes nicht bearbeiten"" auf Seite E-10

Verwendung der Funktion "Anhalten bei Spindelbefehlen"

Aktivieren Sie die Auswahlbox "Anhalten bei Spindelbefehlen", wenn das RS-274D Programm den Bewegungsablauf bei jeder Ausführung eines Spindelbefehls (S, M3, M4 oder M5) anhalten soll. Mit dieser Option kann das Anwenderprogramm die RS-274D Ausführung solange anhalten, bis Ihre Anwenderlogik anzeigt, daß das Spindelprogramm beendet ist. Anschließend kann der Rest des Programms ausgeführt werden. Wird ein Spindelbefehl ausgeführt, dann wird das Bewegungsprogramm solange angehalten und der Merker S.WAIT aktiviert, bis das Anwenderprogramm den Merker S.CONT setzt. Hierdurch wird das Bewegungsprogramm fortgesetzt und die Merker S.WAIT und S.CONT werden von der Steuerung rückgesetzt. Das Anwenderprogramm ist dafür verantwortlich, daß Logik zur Verfügung steht, mit der der Merker S.CONT gesetzt wird, nachdem der Merker S.WAIT aktiviert und der Spindelbefehl beendet wurde.

Beispiel:



Verwendung der Funktion "Anhalten bei Werkzeugwechsel"

Aktivieren Sie die Auswahlbox "Anhalten bei Werkzeugwechsel", wenn das RS-274D Programm den Bewegungsablauf bei jeder Ausführung eines Werkzeugbefehls (T) anhalten soll. Mit dieser Option kann das Anwenderprogramm die RS-274D Ausführung solange anhalten, bis Ihre Anwenderlogik anzeigt, daß der Werkzeugwechsel beendet ist. Anschließend kann der Rest des Programms ausgeführt werden. Wird ein Werkzeugwechsel ausgeführt, dann wird das Bewegungsprogramm solange angehalten und der Merker axis_group.WAIT aktiviert, bis das Anwenderprogramm den Merker axis_group.CONT setzt. Hierdurch wird das Bewegungsprogramm fortgesetzt und die Merker axis_group.WAIT und axis_group.CONT werden von der Steuerung rückgesetzt. Das Anwenderprogramm ist dafür verantwortlich, daß Logik zur Verfügung steht, mit der der Merker axis_group.CONT gesetzt wird, nachdem der Merker axis_group.WAIT aktiviert und der Werkzeugwechsel beendet wurde.

axis_group ist der Name der Achsgruppe, mit der das Werkzeug verknüpft ist. Achsgruppenbezeichnungen werden in der Projektkonfigurationsdatei definiert.

Verwendung von Programmablaufsteuerung in Bewegungsanwendungen

Als Erweiterung zur RS-274D Spezifikation, gestattet die Steuerungssystem-Software die Verwendung der Befehle WHILE und IF-GOTO zusammen mit anderen Motion Control Befehlen, um dadurch die Programmablaufsteuerung zu erweitern. Siehe:

- "Verwendung des Befehls "WHILE"" auf Seite E-19
- "Verwendung des Befehls IF-GOTO" auf Seite E-21

Verwendung des Befehls "WHILE"

Die Befehle in einer WHILE-Schleife werden solange wiederholt ausgeführt, bis ein **< Boolescher Ausdruck >** FALSE wird. Der Programmfluß wird dann mit dem nächsten auf den Befehl END_WHILE folgenden RS-274D Befehl fortgesetzt.

WHILE-Befehlsformat

WHILE (**< Boolescher Ausdruck >)** DO
<RS-274 commands>
END_WHILE

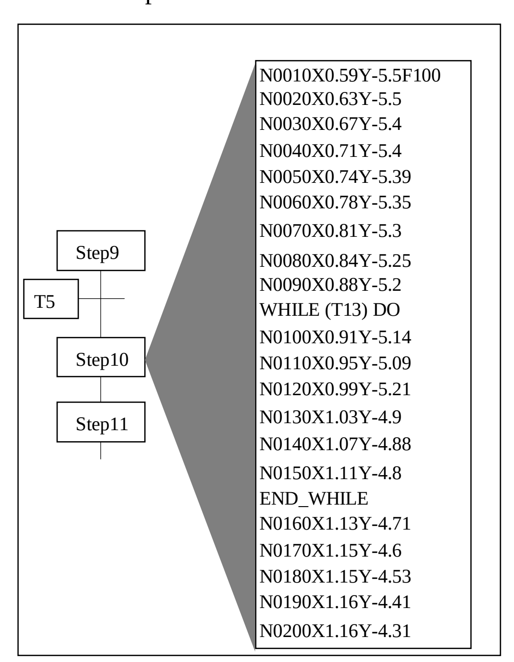
Mit:

< Boolescher Ausdruck > ist ein beliebiger Ausdruck in strukturiertem Text, der einen Booleschen Wert ergibt.

< RS-274 Befehle > bestehen aus normalen RS-274D Befehlen.

Beispiel eines WHILE-Befehls

Nachstehend sehen Sie das Beispiel des WHILE-Befehls:



Ist step10 aktiv und T13 TRUE, dann hält die kontinuierliche Bewegung am WHILE-Befehl an. Die Sätze N0100-N0150 werden bedingt ausgeführt, bis T13 FALSE wird. Dann wird Satz N0160 ausgeführt.

Verwendung des Befehls IF-GOTO

Erreicht der Programmfluß den Befehl IF-GOTO und ist der < Boolesche Ausdruck > TRUE, dann wird der Programmfluß mit dem RS-274D Befehl fortgesetzt, der dem in < Satznummer > enthaltenen Wert entspricht. Ist der < Boolesche Ausdruck > FALSE, dann wird der Programmfluß mit dem auf den Befehl IF-GOTO folgenden RS-274D Befehl fortgesetzt.

Format:

`IF` < Boolescher Ausdruck > `IF-GOTO` < Satznummer >

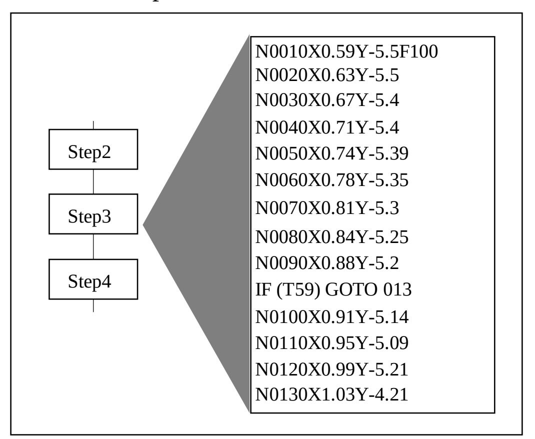
mit:

< Boolescher Ausdruck > ist ein beliebiger Ausdruck in strukturiertem Text, der einen Booleschen Wert ergibt.

< Satznummer > ist die Zeilennummer des nächsten auszuführenden Satzes. Der Wert in < Satznummer > muß genau mit der Zeilennummer des RS-274D Befehls entsprechen, einschließlich aller führenden Nullen und ohne den Bezeichner `N`.

IF-GOTO Beispiel

Nachstehend sehen Sie das Beispiel des IF-GOTO-Befehls:



Ist step3 aktiv, wird der Programmfluß beim Satz N0130 fortgesetzt, wenn T-59 = TRUE. Im anderen Fall werden die Sätze 100-130 ausgeführt.

Verwendung der G56 Makroaufrufe mit Bewegungscode

Als Erweiterung zur RS-274D Spezifikation gestattet die Steuerungssystem-Software, daß Sie Makros (manchmal als Unterprogramme bezeichnet) mit dem Befehl G65 innerhalb des RS-274D Bewegungscodes aufrufen. Ein G65 Unterprogramm wird in zwei einfachen Schritten programmiert:

1. Gestalten Sie Ihre Makros. Siehe "Makro gestalten.”
2. Rufen Sie die Makros zur Ausführung auf. Siehe "Makro zur Ausführung aufrufen.”

Makro gestalten

Sie können das Makro an beliebiger Stelle in Ihrem Programm plazieren. Die Makroaufruffunktion G65 kann vor oder nach dem Makro stehen.

Die Struktur eines Makros unterliegt der folgenden Syntax:

O< x >
< RS-274 Befehle >
M99

mit:

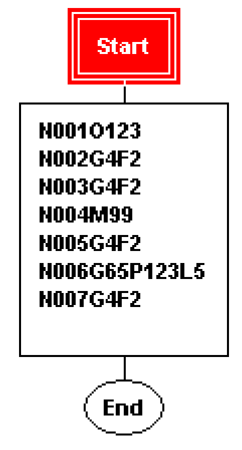
< x > ist ein ganzzahliger Bezeichner für das aufzurufende Makro.

< RS-274 Befehl > ist der Makrocode. Er besteht aus normalen RS-274D Befehlen.

M99 ist ein M-Codesatz, der das Makroende anzeigt.

Die RS-274D Befehle zwischen dem Bezeichner *O*< *x* > und dem M99 Codesatz werden nur ausgeführt, wenn Sie vom G65 Befehl aufgerufen werden. Das heißt, das die Programmbearbeitung den Code zwischen dem O-Wort-Satz und dem M99-Satz überspringt, wenn dieser nicht explizit von einem G65-Satz aufgerufen wurde.

Dies ist ein Beispiel einer Ablaufkette in der Steuerungssystem-Software, die die G65-Funktion veranschaulicht:



Makro zur Ausführung aufrufen

Sie können ein Makro innerhalb eines G65-Satzes aufrufen. Verwenden Sie die folgende Befehlssyntax, um das Makro zur Ausführung aufzurufen:

G65P< marke >L< schleife >

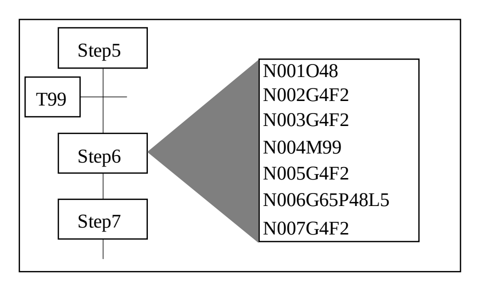
mit:

< marke > ist der ganzzahlige Bezeichner für das Makro. Dieser ganzzahlige Bezeichner muß mit dem Bezeichner in dem O-Wort übereinstimmen.

< schleife > ist eine ganze Zahl die angibt, wie oft das Makro ausgeführt wird.

Der Parameter < marke > muß mit dem Bezeichner *O*< marke > in dem Makro übereinstimmen. Beide Parameters < marke > und <schleife > müssen spezifiziert werden. Soll zum Beispiel Makro 22 einmal aufgerufen werden, dann lautet die Syntax: G65P22L1.

Nachstehend sehen Sie das Beispiel eines Makroaufrufs:



Ist step6 aktiv, überspringt der Programmfluß die ersten vier Sätze, die aus dem Makro bestehen, und beginnt mit der Bearbeitung bei Satz N005. Satz N006 wird dann bearbeitet, wodurch das Makro (Sätze N001-N004) insgesamt fünf Mal aufgerufen wird. Nach der fünften Ausführung des Makros fährt der Programmfluß bei Satz N007 fort.

Motion Control Anwenderprogramme überwachen und ausführen

Dieser Abschnitt beschreibt, wie Sie Tippfeld, Einachsen- und Mehrachsen-Zustandsfeld verwenden und die Achsenkurve überwachen können.

Hinweis: Diese Felder zeigen nur die ersten drei Zeichen eines Achsnamens an. Außerdem sind sie auf die ersten acht Achsen beschränkt.

Verwendung des Tippfeldes

Mit dem Tippfeld können Sie konfigurierte Achsen in Grundstellung bringen und im Tippbetrieb bewegen. Das Achsentippfeld enthält Schaltflächen, über die Sie folgende Punkte auswählen können:

- Tippachse
- Tippart (in Grundstellung bringen, kontinuierlich oder inkremental)
- Tippgeschwindigkeit
- Schrittmaß

Die ausgewählte Achse wird oben am Tippfeld angezeigt und durch Drücken und Halten der Tasten jog+ bzw. jog- in der gewünschten Richtung schrittweise bewegt oder in Grundstellung gebracht. Wird die Taste jog+ bzw. jog- losgelassen, wird die schrittweise Bewegung oder Bewegung zur Grundstellung hin beendet. Bei Inkrementalbewegungen stoppt die Achse, wenn das Schrittmaß zurückgelegt ist. Um ein weiteres Schrittmaß zurückzulegen, müssen Sie die Taste jog+ bzw. jog- erneut drücken und loslassen. Das Tippfeld wird automatisch auf die Anzahl konfigurierter Achsen eingestellt.

Außerdem zeigt das Tippfeld an:

- Absolutposition
- Befohlene Position
- Nachlauffehler und Geschwindigkeitsstatus der ausgewählten Achse

Eine Achsen-Fehleranzeige zeigt an, wenn bei der ausgewählten Achse ein Fehler aufgetreten ist. Beim Auftreten eines Achsfehlers wird die Anzeige rot und die Fehlertaste wird freigegeben. Bei Fehlerfreiheit ist die Anzeige grün.

Drücken Sie die Fehlertaste, um ausführlichere Informationen zum Achsenfehler zu erhalten. Die Anzeige wird wieder grün, wenn der Achsenfehler gelöscht ist.

Achsenkurve überwachen

Verwenden Sie den Menübefehl Anzeigen\Achsenkurve, um über der Zeitachse aufgetragene Achsendaten anzuzeigen. Dieser Befehl aktiviert ein Statusfeld, in dem für eine ausgewählte Achse ausgewählte Statusinformationen über der Zeit aufgetragen werden. Die ausgewählte Achse und die Statusinformation wird oben im Statusfeld angezeigt.

Um die ausgewählte Achse und/oder die Statusinformation zu wechseln, drücken Sie auf die Taste mit der gewünschten Achse und/oder Statusinformation. Mit den Tasten Vergrößern, Verkleinern, Aufwärts verschieben, Abwärts verschieben, Schneller und Langsamer können Sie die Achsenkurve an der gewünschten Stelle positionieren. Sie können gleichzeitig mehrere Kopien der Achszustandskurve aktivieren.

Verwendung des Einachsen-Zustandsfeldes

Zur Anzeige des Einachsen-Bewegungsstatus verwenden Sie den Menübefehl "Anzeigen/Einzel/Achsenstatus. Das Einachsen-Zustandsfeld zeigt den Status einer einzelnen Achse an. Das Einachsen-Zustandsfeld wird automatisch auf die Anzahl konfigurierter Achsen eingestellt.

Um den Status einer Achse anzuzeigen, wählen Sie die Achse, indem Sie die gewünschte Achsentaste auf dem Feld drücken. Der Titel der ausgewählte Achse wird oben im Feld angezeigt.

Der für die ausgewählte Achse angezeigte Status besteht aus:

- Absolutposition
- Befohlene Position
- Nachlauffehler
- Geschwindigkeit

Eine Achsen-Fehleranzeige zeigt an, wenn bei der ausgewählten Achse ein Fehler aufgetreten ist. Beim Auftreten eines Achsfehlers wird die Anzeige rot und die Fehlertaste wird freigegeben. Bei Fehlerfreiheit ist die Anzeige grün.

Drücken Sie die Fehlertaste, um ausführlichere Informationen zum Achsenfehler zu erhalten. Die Anzeige wird wieder grün, wenn der Achsenfehler gelöscht ist.

Sie können mehrere Kopien des Einachsen-Bewegungsstatus aktivieren, um gleichzeitig den vollständigen Achsenstatus mehrere Achsen zu sehen.

Verwenden des Mehrachsen-Zustandsfeldes

Zur Anzeige des Mehrachsen-Bewegungsstatus verwenden Sie den Menübefehl "Anzeigen/Mehrachsenstatus. Im Mehrachsen-Zustandsfeld wird der angegebene Status aller konfigurierter Achsen gleichzeitig angezeigt. Der spezifische Status, der angezeigt wird, wird aus den folgenden Punkten ausgewählt:

- Absolutposition (POS)
- Befohlene Position (CMD)
- Nachlauffehler (FE)
- Geschwindigkeit (VEL)

Wählen Sie den gewünschten Status, indem Sie die gewünschte Statustaste auf dem Feld drücken. Der Titel des ausgewählten Status wird oben im Feld angezeigt.

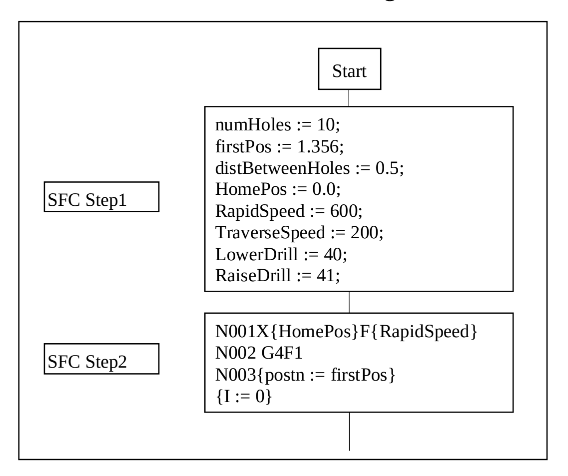
Für jede Achse zeigt eine Achsen-Fehleranzeige an, wenn bei dieser Achse ein Fehler aufgetreten ist. Bei Fehlerfreiheit ist die Anzeige grün. Beim Auftreten eines Achsfehlers wird die Anzeige für diese Achse rot und die Fehlertaste für diese Achse wird freigegeben.

Drücken Sie die Fehlertaste, um ausführlichere Informationen zum Achsenfehler zu erhalten. Die Anzeige wird wieder grün, wenn der Achsenfehler gelöscht ist. Das Mehrachsen-Zustandsfeld wird automatisch auf die Anzahl konfigurierter Achsen eingestellt.

Sie können gleichzeitig mehrere Kopien der Mehrachsen-Bewegungszustands aktivieren.

Strukturierten Text in Motion Control Code einbetten

Als Erweiterung zur IEC-1131-3 Spezifikation gestattet die Steuerungssystem-Software das Einbetten von Zuweisungen und Ausdrücken aus der strukturierten Textsprache in Motion Control Code. Im Beispiel in der nachstehenden Abbildung bewirkt Satz 1 von Step2, daß sich die X-Achse mit der Geschwindigkeit *RapidSpeed* nach *HomePos* bewegt. Die Symbole *HomePos* und *RapidSpeed* sind Werte, die in Step1 von Zuweisungsbefehlen in strukturiertem Text zugewiesen wurden. Die Sätze 3 und 4 von Step2 zeigen Beispiele von Zuweisungsbefehlen in strukturiertem Text, die in Motion Control Code eingebettet wurden.



Beachten Sie folgende Richtlinien, wenn Sie strukturierten Text in einen Motion Control Schritt einbetten:

- Schließen Sie alle eingebetteten strukturierte Textcodeteile in geschweifte Klammern { } ein (siehe Sätze 1, 3 und 4 von Step2 in der vorstehenden Abbildung).
- In einem Satz mit eingebettetem strukturierten Text ist der Befehl N optional.
- Das Semikolon, mit dem normalerweise die einzelnen Codezeilen im strukturierten Text abgeschlossen werden, ist ebenfalls optional.
- Mit Ausnahme der Befehle G und N, können Sie strukturierten Text als einen Parameter für alle Motion Control Befehle verwenden.
- Zum Beispiel ist N016 G4F{pause_time} ein gültiger Satz. pause_time ist ein strukturiertes Textsymbol, das eine Zeitdauer für den Befehl F enthält.
- N100 G{prep_command} ist nicht zulässig. Das strukturierte Textsymbol prep_command kann nicht den Parameter für den Befehl G enthalten.

- Innerhalb der strukturierten Textausdrücke können Sie lokale oder globale Symbole verwenden.. Dies können einfache Symbole oder Feldelemente sein. Sie können Werte im Programm selbst (Step1 in der Abbildung) einstellen oder mit dem dynamischen Datenaustausch (DDE) herunterladen.
- Sie können Zuweisungsanweisungen in getrennten Sätzen verwenden (Step2, Sätze 3 und 4 in der Abbildung).
- Zur Ausgabe von Bewegungsbefehlen aus einem strukturierten Textschritt gibt es drei Funktionsblöcke in strukturiertem Text : AXSJOG, MOVEAXS, und STOPJOG (siehe Seite E-31).

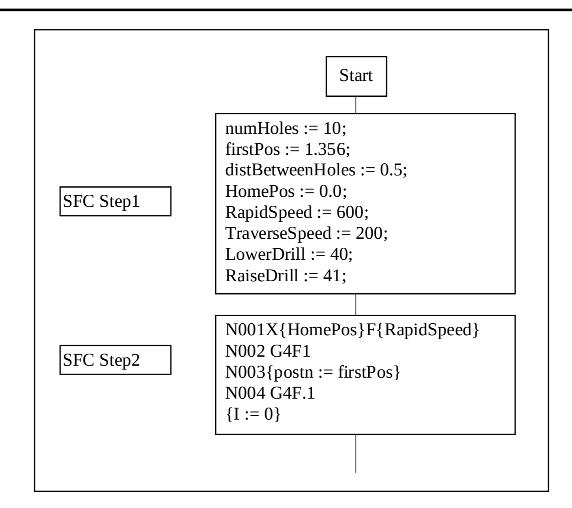
Auswertung des eingebetteten strukturierten Textcodes

Trifft der Programmfluß auf einen Block mit strukturiertem Text, werden Programmausführung und alle Bewegungsaktivitäten kurz angehalten. Der Programmfluß wird mit den nächsten auf den strukturierten Text folgenden Satz fortgesetzt.

Die Ausdrücke werden zum Zeitpunkt der Ausführung ausgewertet (wenn der Satz bearbeitet wird), sie können das Ergebnis von Echtzeiteingangssignalen (z.B. Sensordaten) darstellen. Sie werden als ganze oder reelle Zahlen entsprechend dem Motion Control Wort ausgewertet.

Da die Steuerungssystem-Software über eine "Vorauslesefunktion” verfügt, werden Zuweisungsbefehle in strukturiertem Text gelesen und ausgeführt, ehe der Programmfluß den Satz antrifft, in dem sie enthalten sind. Sie können die Ausführung einer strukturierten Textzuweisung steuern, indem Sie einen Befehl G04 in den Satz einbringen, der vor dem Satz mit der Zuweisung steht. Die "Vorauslesefunktion” geht nicht über einen Befehl G04 hinaus.

Im Beispiel in der nachstehenden Abbildung wird der Zuweisungsbefehl im fünften Satz von Step2 erst ausgeführt, nachdem der Programmfluß auf die Anweisung getroffen ist. Ohne den Befehl G04 im vierten Block könnte das Symbol I auf Null gesetzt werden, ehe der zweite oder dritte Satz ausgeführt wurde.


Start
SFC Step1
numHoles := 10;
firstPos := 1.356;
distBetweenHoles := 0.5;
HomePos := 0.0;
RapidSpeed := 600;
TraverseSpeed := 200;
LowerDrill := 40;
RaiseDrill := 41;
SFC Step2
N001X{HomePos}F{RapidSpeed}
N002 G4F1
N003{postn := firstPos}
N004 G4F.1
{I := 0}

Bewegungsfunktionen in strukturiertem Text

Die folgenden Bewegungsfunktionen in strukturiertem Text stehen zur Verfügung: AXSJOG, MOVEAXS und STOPJOG.

AXSJOG

Die Funktion AXSJOG startet einen Tippbefehl an einer angegebenen Achse.

Format:

AXSJOG(**<achse>**, JOGDIR:= **<richtung>** , JOGTYPE:= **<typ>**, JOGSPD:= **<geschwindigkeit>**, JOGDIST:= **<entfernung>**);

mit:

<achse> ist die Achsbezeichnung (`X, Y,Z, usw.`)

<richtung> ist die Tipprichtung (`JOGPLUS` oder `JOGMINUS`)

<typ> ist die Tippart (`JOGCONT` für kontinuierliches Tippen, `JOGINCR` für inkrementelles Tippen, oder `JOGHOME` für eine Grundstellung

<geschwindigkeit> und **<entfernung>** sind reelle Zahlen, Symbole oder Ausdrücke, die zu reellen Zahlen werden und Tippgeschwindigkeit und Entfernung angeben.

Beispiel:

```
AXSJOG (X, JOGDIR:=JOGMINUS, JOGTYPE:=JOGCONT, JOGSPD:=10.0);
```

MOVEAXS

Die Funktion MOVEAXS startet einen Verfahrbefehl an einer angegebenen Achse.

Format:

`AXSJOG(`<achse>`, POSTN:=` <position> `, VEL:=` <geschwindigkeit>`,`
`ACCEL:=` <beschleunigung>`);`

mit:

<achse> ist die Achsbezeichnung (`X, Y,Z, usw.`)

<position>, **<geschwindigkeit>**, und **<beschleunigung>** sind reelle Zahlen, Symbole, oder Ausdrücke, die reelle Zahlen ergeben und Verfahrposition, Geschwindigkeit und Beschleunigung angeben.

Beispiel:

```
MOVEAXS (X, POSTN:= positionA, VEL:= velocityA, ACCEL:=10.0);
```

STOPJOG

Die Funktion STOPJOG stoppt einen Tippbefehl an einer angegebenen Achse.

Format:

STOPJOG(**<achse>**);

mit:

<achse> ist die Achsbezeichnung (`X, Y`, `Z, usw.`).

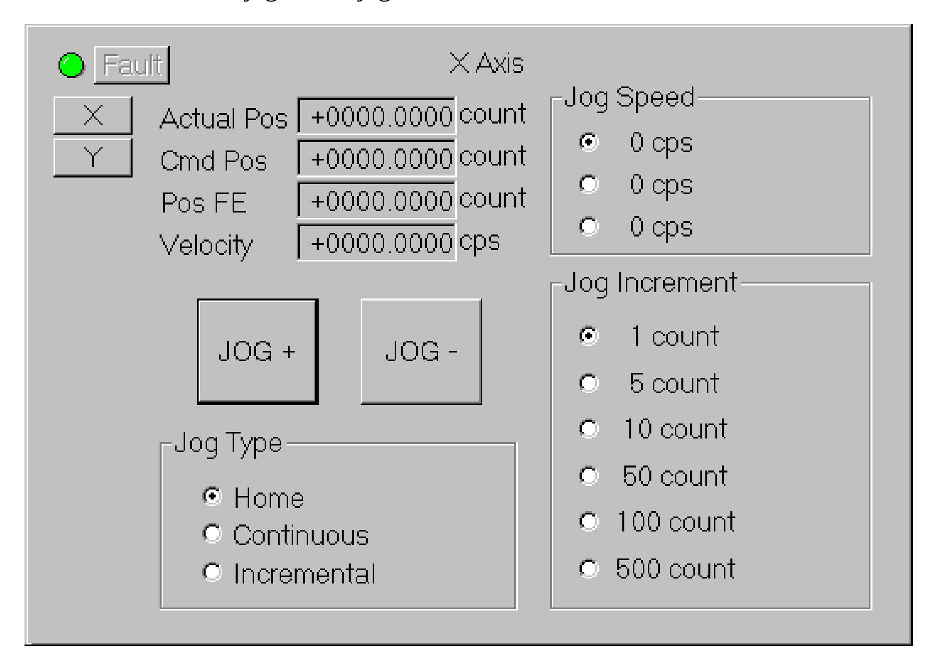
Beispiel:

```
STOPJOG (Z);
```

Bedienerschnittstellen-Bewegungssteuerungen

Tippfeld

Das Tippfeld liefert die Funktionen, die benötigt werden, um konfigurierte Achsen in Grundstellung zu bringen und im Tippbetrieb zu bewegen. Es enthält Schaltflächen, mit denen Tippachse, Tippart (Grundstellung, kontinuierlich oder inkremental), Tippgeschwindigkeit und Schrittmaß eingestellt werden können. Die ausgewählte Achse wird oben im Tippfeld angezeigt. Das Tippfeld zeigt auch Absolutposition, Sollposition, Nachlauffehler und Geschwindigkeitsstatus der ausgewählten Achse an. Mit einer Achsenfehleranzeige wird angezeigt, daß bei der eingestellten Achse ein Fehler aufgetreten ist. Bei Fehlerfreiheit ist die Anzeige grün. Beim Auftreten eines Achsfehlers wird die Anzeige rot und die Fehlertaste wird freigegeben. Drücken Sie die Fehlertaste, um ausführlichere Informationen zum Achsenfehler zu erhalten. Die Anzeige wird wieder grün, wenn der Achsenfehler gelöscht ist. Die ausgewählte Achse wird durch Drücken und Halten der Tasten jog+ bzw. jog- in der gewünschten Richtung schrittweise bewegt oder in Grundstellung gebracht. Wird die Taste jog+ bzw. jog- losgelassen, wird die schrittweise Bewegung oder Bewegung zur Grundstellung hin abgebrochen. Bei Inkrementalbewegungen stoppt die Achse, wenn das Schrittmaß zurückgelegt ist. Um ein weiteres Schrittmaß zurückzulegen, müssen Sie die Taste jog+ bzw. jog- erneut drücken und loslassen.

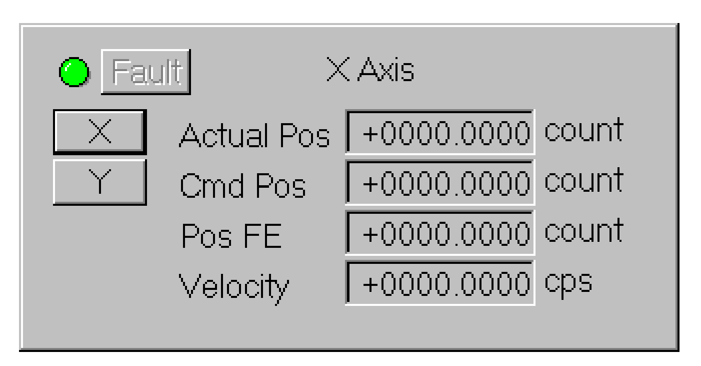


Das Tippfeld wird automatisch auf die Anzahl konfigurierter Achsen eingestellt. Für das Tippfeld-Bedienelement gibt es keine Konfigurationsoptionen.

Einachsen-Zustandsfeld

Das Einachsen-Zustandsfeld zeigt den Status einer einzelnen Achse an. Wählen Sie die Achse, deren Zustand angezeigt werden soll, indem Sie die gewünschte Achsentaste auf dem Feld drücken. Der Titel der ausgewählte Achse wird oben im Feld angezeigt. Der für die ausgewählte Achse angezeigte Status besteht aus Absolutposition, Sollposition, Nachlauffehler und Geschwindigkeit. Mit einer Achsenfehleranzeige wird angezeigt, daß bei der eingestellten Achse ein Fehler aufgetreten ist. Bei Fehlerfreiheit ist die Anzeige grün. Beim Auftreten eines Achsfehlers wird die Anzeige rot und die Fehlertaste wird freigegeben. Drücken Sie die Fehlertaste, um ausführlichere Informationen zum Achsenfehler zu erhalten. Die Anzeige wird wieder grün, wenn der Achsenfehler gelöscht ist.

Das Einachsen-Zustandsfeld wird automatisch auf die Anzahl konfigurierter Achsen eingestellt. Für das Einachsen-Zustandsfeld-Bedienelement gibt es keine Konfigurationsoptionen.



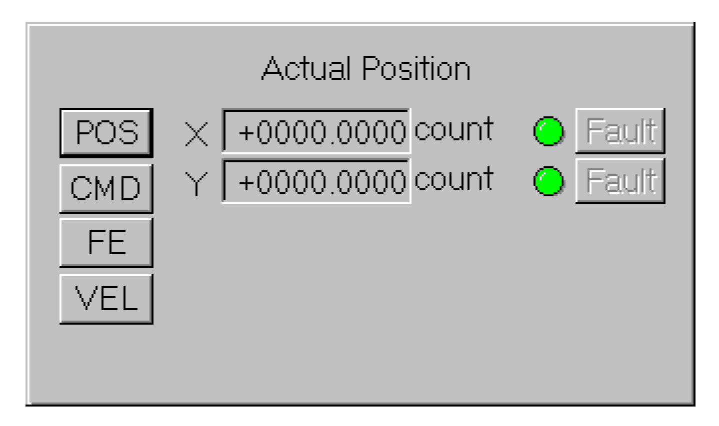
Verwendung des Einachsen-Zustandsfeldes

Die ausgewählte Achse wird oben im Statusfeld angezeigt. Um die ausgewählte Achse zu wechseln, drücken Sie auf die Schaltfläche mit der gewünschten Achsbezeichnung. Sie können mehrere Kopien des Einachsen-Bewegungsstatus aktivieren, um gleichzeitig den vollständigen Achsenstatus mehrere Achsen zu sehen.

Mehrachsen-Zustandsfeld

Im Mehrachsen-Zustandsfeld wird der angegebene Status aller konfigurierter Achsen gleichzeitig angezeigt. Der spezifische Status, der angezeigt wird, wird aus den folgenden Punkten ausgewählt: Absolutposition (POS), Sollposition (CMD), Nachlauffehler (FE) und Geschwindigkeit (VEL). Wählen Sie den gewünschten Status, indem Sie die gewünschte Statustaste auf dem Feld drücken. Der Titel des ausgewählten Status wird oben im Feld angezeigt. Für jede Achse wird mit einer Achsenfehleranzeige angezeigt, daß bei der eingestellten Achse ein Fehler aufgetreten ist. Bei Fehlerfreiheit ist die Anzeige grün. Beim Auftreten eines Achsfehlers wird die Anzeige für diese Achse rot und die Fehlertaste für diese Achse wird freigegeben. Drücken Sie die Fehlertaste, um ausführlichere Informationen zum Achsenfehler zu erhalten. Die Anzeige wird wieder grün, wenn der Achsenfehler gelöscht ist.

Das Mehrachsen-Zustandsfeld wird automatisch auf die Anzahl konfigurierter Achsen eingestellt. Für das Mehrachsen-Zustandsfeld-Bedienelement gibt es keine Konfigurationsoptionen.



Verwendung des Mehrachsen-Zustandsfeldes

Die ausgewählte Statusinformation wird oben im Statusfeld angezeigt. Um die ausgewählte Statusinformation zu wechseln, drücken Sie auf die Taste mit der gewünschten Statusinformation. Sie können gleichzeitig mehrere Kopien der Mehrachsen-Bewegungszustands aktivieren.

RS274 Satzanzeige

Mit der RS274 Satzanzeige können Sie die aktiven RS274 Satzbefehle im Bedienerschnittstellenmenü anzeigen. Es werden alle Satzbefehle im aktiven Ablaufketten-Schritt angezeigt. Der aktive Satz wird grün hervorgehoben und am Anfang des Satzes steht ein Zeiger (>).

Bearbeiten einer RS274 Satzanzeige

Um eine RS274 Satzanzeige zu bearbeiten doppelklicken Sie auf die gewünschte RS274 Satzanzeige oder selektieren die gewünschte RS274 Satzanzeige und drücken dann auf Enter. Hierauf wird die Dialogbox "RS274 Satzanzeige bearbeiten" angezeigt und gestattet Ihnen, die RS274 Satzanzeigen-Definitionsdaten zu verändern.

Mit der RS274 Programmsymbol-Kombinationsbox können Sie das STRING-Symbol definieren, das den Dateipfad der Ablaufkette enthält, aus der aktive RS274 Sätze angezeigt werden. Zur Auswahl eines Anzeigesymbols für das RS274 Programmsymbol wählen Sie entweder einen Symbolnamen aus dem Drop-Down-Listenfeld oder geben einen Symbolnamen in das Editierfeld ein.

Die RS274 Satzanzeige verwendet OLE-Technik zur Anbindung an den Programmeditor. Der Programmeditor zeigt die RS274 Sätze in der RS274 Satzanzeige auf dem Bedienermenü an. Bei jeder Änderung des Programmsymbols erstellt die RS274 Satzanzeige die Verbindung zu der neuen Ablaufkette.

Es werden nur RS274 Sätze angezeigt, die in der obersten Ablaufkette liegen. RS274 Sätze in einer Makroschritt-Ablaufkette werden nicht angezeigt.

Leistungsbetrachtungen

Zur Verknüpfung der RS274 Satzanzeige mit dem Programmeditor muß das OLE-Subsystem die Programmeditor-Anwendung ausführen. Die zum Hochlauf des Programmeditors benötigte Zeit kann in der Bedienerschnittstelle einiger Systeme zu unerwünschten Verzögerungen führen. Um diese Verzögerung in der Bedienerschnittstelle zu eliminieren werden die beiden folgenden Empfehlungen ausgesprochen:

1) Starten Sie den Programmeditor minimiert im Hintergrund (falls gewünscht nur mit Lesezugriff) vor der Bedienerschnittstelle.

2) Initialisieren Sie alle RS274 Satzanzeigesymbole mit den gewünschten AS-Dateipfaden. Da alle OLE-Verknüpfungen bei der Aktivierung der Bedienerschnittstelle angeschlossen sind, tritt die Verzögerung durch die Verknüpfung beim Systemstart auf.

Anhang F — Pointer in strukturiertem Text

Vorsicht

Pointer sollten nur von Fachleuten eingesetzt werden. Falsche Verwendung kann zu unvorhersehbarem Betriebsverhalten und großen Problemen beim Austesten führen.

Strukturierter Text besitzt zwei Pointeroperatoren: Den Pointeroperator Referenzierung (&) und den Pointeroperator Dereferenzierung(*). Diese Operatoren werden zur *indirekten Adressierung* eingesetzt.

In diesem Anhang finden Sie Informationen zu folgenden Themen:

- Adressierung
- Pointeroperatoren
- Pointersymboldefinition
- Feldpointer

Adressierung

Bei der *direkten Adressierung* wird einem Datenwert eine symbolische Bezeichnung zugewiesen. Mit dieser symbolischen Bezeichnung wird direkt auf den Datenwert zugegriffen. Im nachstehenden Beispiel wird X der Wert des Datenwertes Y zugewiesen.

```
X := Y;
```

Bei der *indirekten Adressierung* weist eine symbolische Bezeichnung (allgemein *Pointersymbol* genannt) auf die Adresse, unter der der Datenwert gespeichert ist. Um den aktuellen Datenwert über indirekte Adressierung zu erhalten, wird zunächst über das Pointersymbol die Adresse des Datenwerts ermittelt, und dann über diese Adresse der aktuelle Datenwert geholt.

In den nachstehenden Beispielen ist pVar1 ein Pointersymbol.

pVar1 := & X;	pVar1 ist der Adresse des Datenwertes X zugewiesen.
Y := *pVar1;	Y ist der in X enthaltene Wert zugewiesen, da pVar1 die Adresse von X enthält.
*pVar1 := Y;	X ist der in Y enthaltene Wert zugewiesen.

Pointer-Operatoren

Wir nehmen an, daß VarInt1 und VarInt2 ganzzahlige Symbole sind und daß pInt als ein Pointer auf eine ganze Zahl definiert ist.

Eine Adresse einem Pointersymbol zuweisen

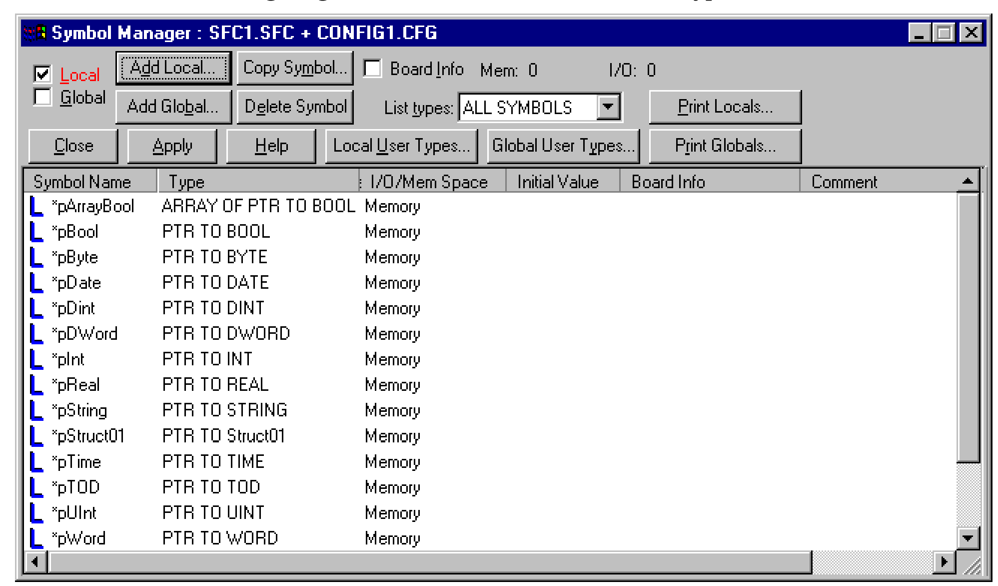
pInt := & VarInt1;	pInt wird der Adresse von VarInt1 zugewiesen. "&" bedeutet "erhalte die Adresse von”

Einen durch ein Pointersymbol adressierten Datenwert erhalten

VarInt2 := * pInt;	VarInt2 wird dem Wert von VarInt1 zugewiesen. * bedeutet "erhalte den Wert aus”.

Pointersymboldefinition

Wie andere Symbole auch werden Pointersymbol im Symbolmanager definiert. Die nachstehende Abbildung zeigt die Definition mehrerer Pointertypen.



Pointerhinweise

1. Wird ein Pointersymbol definiert, dann wird es als ein Pointer auf ein Symbol eines bestimmten Datentyps (REAL, INT, STRING, usw.) definiert. Das Pointersymbol pVar1 könnte zum Beispiel der Adresse eines beliebigen Symbols seines Datentyps zugewiesen werden.

2. Sie können Pointer auf Standard-Datentypen, Anwenderstrukturen oder Felder definieren. Pointer auf Funktionsblöcke oder Systemobjekte *können nicht* definiert werden.

3. Ein Pointersymbol kann durch ein Symbol des gleichen Grundtyps ersetzt werden, indem ihm der Dereferenzierungsoperator * vorangestellt wird. (*pVarInt1 = VarInt1, wenn die Zuweisung pVarInt1:=&VarInt1 erfolgte.)

4. Ein Pointer auf eine Struktur kann direkt anstelle der Strukturbezeichnung verwendet werden. Zum Beispiel:

   Wir nehmen an, daß eine Anwenderstruktur UserStruct1 definiert wurde und daß pStruct1 als ein Pointer auf diese Anwenderstruktur definiert wurde.

   Bei Anwenderstruktursymbolen ist der Name der Anwenderstruktur (hier: UserStruct1) ein Pointer auf die Datenwerte der Anwenderstruktur. Der Name der Anwenderstruktur kann jedoch nicht einer anderen Adresse zugewiesen werden, er zeigt immer auf die Anwenderstruktur.

pStruct1 := UserStruct1;	pStruct1 ist der Adresse von UserStruct1 zugewiesen.
pStruct1.intMember1 := VarInt1;	Das Element intMember1 von UserStruct1 ist dem Wert von VarInt1 zugewiesen.

5. Ein Pointer, der einer Adresse in einem Feld (z.B. pInt:= &intArray[5];) zugewiesen ist, kann mit dem Feldindizierungsoperator (pInt[Index]) zur Indizierung in das Feld verwendet werden. Dieser Index beginnt mit 0 (dem Feldelement, auf das der Pointer zeigt) und geht weiter bis zum Ende des Feldes. Zum Beispiel:

   intArray ist definiert als ein Feld mit zehn ganzzahligen Werten (ARRAY[1..10])
   pInt ist definiert als ein Pointer auf einen ganzzahligen Typ
   Somit:

   ```
   pInt:= &intArray[5];
   pInt[0]:= 0;          (*intArray[5]*)
   pInt[5]:= 5;          (*intArray[10]*)
   ```

   Weitere Beispiele zur Verwendung von Feldpointern finden Sie auf Seite E-5.

6. Einem Pointersymbol können Sie wie folgt einen Nullwert zuweisen:

pInt := NULL;	pInt wird dem NULL Pointerwert zugewiesen. Pointersymbole können NULL zugewiesen und mit diesem verglichen (gleich oder ungleich) werden. NULL ist ein System-Schlüsselwort.

7. Die Adresse eines Pointersymbols kann in einem MOVE-Funktionsblock initialisiert werden. Der Ausgang des MOVE-Funktionsblocks ist das Pointersymbol, das initialisiert werden soll. Der Eingang zum MOVE-Funktionsblock ist entweder ein anderes Pointersymbol oder ein Direktsymbol, vor dem der Pointer-Referenzierungsoperator & steht.

8. Pointer *können nicht* in FILE-Funktionen, Bitfeldfunktionen (SHL, AND_BITS usw.) oder STRING_TO_ARRAY-Funktionen weitergegeben werden.

Feldpointer

Verwendung eines Feldpointers

Wir nehmen an, daß IntArray ein Feld mit zehn ganzen Zahlen (IntArray: ARRAY [1..10] OF INT) ist und pArray als Pointer auf ganze Zahlen definiert ist.

Bei Feldsymbolen ist der Name des Feldes (hier: IntArray) ähnlich einem Pointer auf die Datenwerte des Feldes. Der Feldname kann jedoch nicht einer anderen Adresse zugewiesen werden, er zeigt immer auf das Feld.

pArray := &IntArray;	pArray ist die Adresse von IntArray zugewiesen. Er zeigt auf das erste Feldelement.
pArray[index] := VarInt1;	IntArray[index] ist der Wert von VarInt1 zugewiesen.
pArray := & IntArray[index];	pArray zeigt auf das Feldelement beim Index.

Einen Pointer auf ein Feld mit Anwenderstrukturen anwenden

Wir nehmen an, daß ein Feld mit Anwenderstrukturen structArray (structArray : ARRAY [1..10] OF USER_STRUCT1) sowie ein Pointer zu der Anwenderstruktur pStructArray definiert wurden (pStructArray : PTR TO USER_STRUCT1).

pStructArray := &structArray;	pStructArray ist der Adresse von structArray zugewiesen.
pStructArray[5].intMember1 := VarInt1;	Das Element intMember1 von structArray[5] ist dem Wert von VarInt1 zugewiesen.

Ein Pointerfeld verwenden, das auf einen INT zeigt

Wir nehmen an, daß ein Pointerfeld ArrayPtr (ARRAY [1..10] OF PTR TO INT) definiert ist.

ArrayPtr[index] := &VarInt1;	ArrayPtr[index] ist der Adresse von VarInt1 zugewiesen.
*ArrayPtr[index] := VarInt2;	VarInt1 ist dem Wert von VarInt2 zugewiesen.

Ein Pointerfeld verwenden, das auf ein INT-Feld zeigt

Wir nehmen an, daß ein Feld ganzzahliger Werte intArray (ARRAY [1..10] OF INT) sowie ein Feld mit Pointern ArrayPtr (ARRAY [1..10] OF PTR TO INT) definiert wurden.

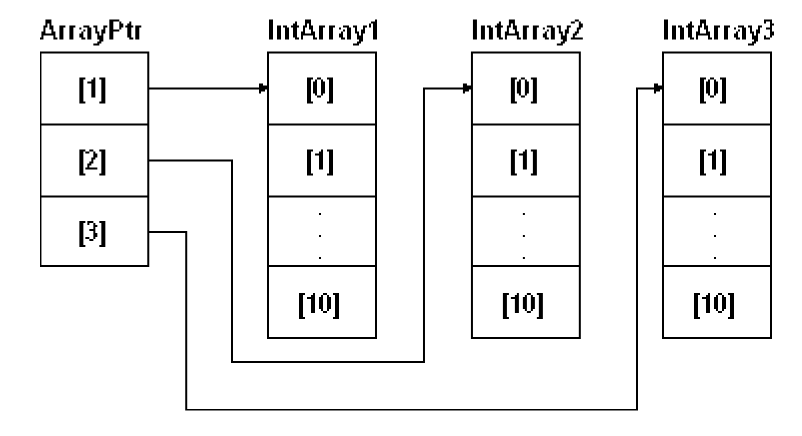
ArrayPtr[index] := &IntArray;	ArrayPtr[index] ist die Adresse des ersten Elements von intArray zugewiesen.
*ArrayPtr[index] := VarInt1;	IntArray[1] ist der Wert von VarInt1 zugewiesen.
ArrayPtr[index][1] := VarInt1;	Gleiche Zuweisung wie oben.

Ein Pointerfeld verwenden, das auf ein INT-Feld zeigt

Wir nehmen an, daß ein Feld ganzzahliger Werte intArray1, intArray2, IntArray3 (ARRAY [0..10] OF INT) sowie ein Feld mit Pointern ArrayPtr (ARRAY [1..3] OF PTR TO INT) definiert wurden.

```
ArrayPtr[1] := &IntArray1[0];
ArrayPtr[2] := &IntArray2[0];
ArrayPtr[3] := &IntArray3[0];
```

Siehe nachstehende Abbildung:



Es kann folgende Syntax verwendet werden:

```
ArrayPtr[1] [0] := VarInt1;     (*IntArray1[0] = VarInt1*)
ArrayPtr[1] [1] := VarInt2;     (*IntArray1[1] = VarInt2*)
ArrayPtr[1] [2] := VarInt3;     (*IntArray1[2] = VarInt3*)

ArrayPtr[2] [0] := VarInt1;     (*IntArray2[0] = VarInt1*)
ArrayPtr[2] [1] := VarInt2;     (*IntArray2[1] = VarInt2*)
ArrayPtr[2] [2] := VarInt3;     (*IntArray2[2] = VarInt3*)

ArrayPtr[3] [0] := VarInt1;     (*IntArray3[0] = VarInt1*)
ArrayPtr[3] [1] := VarInt2;     (*IntArray3[1] = VarInt2*)
ArrayPtr[3] [2] := VarInt3;     (*IntArray3[2] = VarInt3*)
```

Anhang G

Beschreibung der Befehlssätze

Dieser Anhang faßt die Befehlssätze von Kontaktplan, Ablaufsprache, strukturiertem Text und Anweisungsliste zusammen. Ausführliche Erläuterungen der einzelnen Anweisungen finden Sie in der Online-Hilfe.

Kontaktplan-Befehlssatz - Zusammenfassung

Kontakte und Spulen

Typ	**Schaltfläche**	**Beschreibung**
Schließerkontakt (NO)	-] [-	Der Kontakt gibt Stromfluß weiter, wenn der von ihm dargestellte Punkt TRUE ist.
Öffnerkontakt (NC)	-]/[-	Der Kontakt gibt Stromfluß weiter, wenn der von ihm dargestellte Punkt FALSE ist.
Kontakt zur Erkennung positiver Übergänge	-]P[-	Der Kontakt gibt Stromfluß weiter, wenn der von ihm dargestellte Punkt unmittelbar vor der Auswertung des Kontaktes von FALSE nach TRUE wechselt.
Kontakt zur Erkennung negativer Übergänge	-]N[-	Der Kontakt gibt Stromfluß weiter, wenn der von ihm dargestellte Punkt unmittelbar vor der Auswertung des Kontaktes von TRUE nach FALSE wechselt.
Ausgangsspule	-( )-	Die Spule setzt den von ihm dargestellten Punkt auf TRUE, wenn sie Stromfluß erhält.
Negierte Ausgangsspulen	-(/)-	Die Spule setzt den von ihm dargestellten Punkt auf TRUE, wenn sie keinen Stromfluß erhält.
SETZE-Spule (verriegeln)	-(S)-	Die Spule setzt den von ihm dargestellten Punkt auf TRUE, wenn sie Stromfluß erhält. Sie kann nur durch eine RÜCKSETZE-Spule auf FALSE gesetzt werden.
RÜCKSETZE-Spule (entriegeln)	-(R)-	Die Spule setzt den von ihm dargestellten Punkt auf FALSE, wenn sie Stromfluß erhält. Der Punkt bleibt FALSE (das Objekt ist rückgesetzt oder entriegelt), selbst wenn die Spule keinen Stromfluß mehr hat. Er kann nur durch eine SETZE-Spule auf TRUE gesetzt werden.

Typ	Schaltfläche	Beschreibung
Spule zur Erkennung positiver Übergänge	-(P)-	Wechselt der Stromfluß zur Spule von FALSE auf TRUE, dann setzt die Spule den von ihr dargestellten Punkt auf TRUE. Der Punkt bleibt für die Dauer eines Zyklus TRUE, wenn er nicht an anderer Stelle auf FALSE gesetzt wird.
Spule zur Erkennung negativer Übergänge	-(N)-	Wechselt der Stromfluß zur Spule von TRUE auf FALSE, dann setzt die Spule den von ihr dargestellten Punkt auf TRUE. Das Objekt bleibt für die Dauer eines Zyklus TRUE, wenn es nicht an anderer Stelle auf FALSE gesetzt wird.
Sprungmerker/Marke	(JMP) [LBL]	Mit den Elementen Sprung und Marke können Sie Abschnitte des Programmcodes vorübergehend deaktivieren. Sprungmerker und Marke müssen immer zusammen verwendet werden. Eine Marke ohne Sprungmerker verursacht einen Laufzeitfehler.
Ablaufketten-Transitionsmerker	-(TRANS)-	Der Ablaufketten-Transitionsmerker ist ein Kontaktplanprogrammelement, das Sie nur unter bestimmten Bedingungen (innerhalb einer Ablaufplanaktion) in einer Ablaufkette verwenden können.

Bitfolge

Typ	Abk.	Beschreibung
Logisch UND	AND	Erzeugt eine bitweise UND-Verknüpfung von zwei Zahlen.
Logisch NICHT	NOT	Erzeugt ein bitweises Komplement einer Zahl.
Logisch ODER	OR	Erzeugt eine bitweise ODER-Verknüpfung von zwei Zahlen.
Nach links rotieren	ROL	Rotiert den Eingangswert nach links um die durch die Schiebezahl spezifizierte Anzahl Bits.
Nach rechts rotieren	ROR	Rotiert den Eingangswert nach rechts um die durch die Schiebezahl spezifizierte Anzahl Bits.
Nach links verschieben	SHL	Verschiebt den Eingangswert nach links um die durch die Schiebezahl spezifizierte Anzahl Bits.
Nach rechts verschieben	SHR	Verschiebt den Eingangswert nach rechts um die durch die Schiebezahl spezifizierte Anzahl Bits.
Logisches Exklusiv-ODER	XOR	Erzeugt eine bitweise Antivalenz von zwei Zahlen.

Zeichenfolge

Typ	Abk.	Beschreibung
Ketten	CAT	Hängt die zweite Eingangs-Zeichenfolge an das Ende der ersten Eingangs-Zeichenfolge an.
Löschen	DEL	Löscht Zeichen aus der Eingangs-Zeichenfolge heraus.
Suchen	FIND	Sucht in einer Eingangs-Zeichenfolge nach einer anderen.
Einfügen	INS	Fügt eine Eingangs-Zeichenfolge in eine andere ein.
Links kopieren	LEFT	Kopiert die am weitesten links stehenden Zeichen der Eingangs-Zeichenfolge.
Länge speichern	LEN	Speichert die Länge der Eingangs-Zeichenfolge.
Mitte kopieren	MID	Kopiert Zeichen aus der Mitte der Eingangs-Zeichenfolge.
Rechts kopieren	RGHT	Kopiert die am weitesten rechts stehenden Zeichen der Eingangs-Zeichenfolge.
Ersetzen	RPLC	Ersetzt Zeichen in einer Eingangs-Zeichenfolge durch eine andere Eingangs-Zeichenfolge.
Bytefeld-in-Zeichenfolge	BATOS	Nimmt das Eingangsbytefeld und speichert die Bytes als Zeichen in einer Zeichenfolge.
Umwandlung ganze Zahl in ASCII	ITOA	Wandelt den ganzzahligen Eingangswert in eine ASCII-Zeichenfolge um.
Anzeige einer Meldung im Bedienermenü	MSGB	Zeigt ein Meldungsfeld im Bedienermenü an.
Meldungsanzeige in einem Ausgabefenster	MSGW	Zeigt eine Meldung im Ausgabefenster an.
Umwandlung REAL in ASCII	RTOA	Wandelt den reellen Eingangswert in eine ASCII-Zeichenfolge um.
Zeichenfolge-in-Bytefeld	STOBA	Nimmt die Eingangs-Zeichenfolge und speichert die Zeichen der Zeichenfolge in ein Bytefeld.

Vergleich (Zeichenfolge+)

Typ	Abk.	Beschreibung
Gleich	EQ	Prüft zwei Eingangswerte auf Gleichheit.
Größer als oder gleich	GE	Prüft, ob der erste Eingangswert größer als der zweite Eingangswert oder gleich diesem ist.
Größer als	GT	Prüft, ob der erste Eingangswert größer als der zweite Eingangswert ist.
Kleiner als oder gleich	LE	Prüft, ob der erste Eingangswert kleiner als der zweite Eingangswert oder gleich diesem ist.
Kleiner als	LT	Prüft, ob der erste Eingangswert kleiner als der zweite Eingangswert ist.
Ungleich	NE	Prüft zwei Eingangswerte auf Ungleichheit.

Umwandlung

Typ	Abk.	Beschreibung
Bytefeld-in-Zeichenfolge	BATOS	Konvertiert ein Bytefeld in eine Zeichenfolge.
Datum in Zeichenfolge	DateToString	Konvertiert ein Datum in eine Zeichenfolge.
Ganze Zahl in Zeichenfolge	ITOA	Konvertiert eine ganze Zahl in eine Zeichenfolge.
REAL in Zeichenfolge	RTOA	Konvertiert eine reelle Zahl in eine Zeichenfolge.
RGB in DWORD	RGB_TO_DWORD	Konvertiert ein Tripel aus Rot-, Grün- und Blauwerten in ein DWORD.
Zeichenfolge-in-Bytefeld	STOBA	Konvertiert eine Zeichenfolge in ein Bytefeld.
Zeichenfolge in Datum	StringToDate	Konvertiert eine Zeichenfolge in ein Datum.
Zeichenfolge in ganze Zahl	ATOI	Konvertiert eine numerische ASCII-Zeichenfolge in einen ganzzahligen Wert.
Zeichenfolge in REAL	ATOR	Konvertiert eine numerische ASCII-Zeichenfolge in einen reellen Wert.
Zeichenfolge in Uhrzeit	StringToTOD	Konvertiert eine Zeichenfolge in eine Uhrzeit.
Uhrzeit in Zeichenfolge	TODToString	Konvertiert eine Uhrzeit in eine Zeichenfolge.

Zähler und Timer

Typ	Abk.	Beschreibung
Abwärts zählen	CTD	Zählt Ereignisse und verringert den Zählerstand dabei um eins..
Aufwärts zählen	CTU	Zählt Ereignisse und erhöht den Zählerstand dabei um eins..
Aufwärts/abwärts zählen	CTUD	Zählt Ereignisse aufwärts oder abwärts.
Abschaltverzögerung	TOF	Bietet eine Verzögerung beim Abschalten von Ereignissen.
Einschaltverzögerung	TON	Bietet eine Verzögerung beim Einschalten von Ereignissen.
Timerimpuls	TP	Wird durch einen Impuls aktiviert und bietet eine Verzögerung beim Abschalten von Ereignissen.

Flankenerkennung

Typ	Abk.	Beschreibung
Triggersignal bei abfallender Flanke	FTRG	Schaltet einen Ausgang durch, wenn durch abfallende Flanke getriggert.
Triggersignal bei steigender Flanke	RTRG	Schaltet einen Ausgang durch, wenn durch steigende Flanke getriggert.

Erweiterte Timer

Erweiterte Timer-Ausschaltverzögerung	XTOF	Bietet eine Verzögerung beim Abschalten von Ereignissen.
Erweiterte Timer-Einschaltverzögerung	XTON	Bietet eine Verzögerung beim Einschalten von Ereignissen.
Erweiterter Timer-Impuls	XTP	Wird durch einen Impuls aktiviert und bietet eine Verzögerung beim Abschalten von Ereignissen.

Datei

Typ	Abk.	Beschreibung
Datei schließen	FCLOS	Schließt eine Datei.
Datei kopieren	FCOPY	Kopiert eine Datei.
Datei löschen	FDEL	Löscht eine Datei.
Datei erstellen	FNEW	Erstellt eine neue Datei.
Datei öffnen	FOPEN	Öffnet eine bestehende Datei.
Datei lesen	FREAD	Liest Daten aus einer Datei.
Datei zurückspulen	FRWN D	Spulte eine Datei an den Anfang zurück.
In Datei schreiben	FWRIT	Schreibt Daten in eine Datei.

Arithmetische Funktionen

Typ	Abk.	Beschreibung
Absolutwert	ABS	Berechnet den Absolutwert einer Zahl.
Addition	ADD	Addiert zwei Zahlen.
Division	DIV	Dividiert eine Zahl durch eine andere.
Exponent	EXPT	Erhebt die erste Zahl zu der durch die zweite Zahl angegebene Potenz.
Modulo	MOD	Dividiert eine Zahl durch eine andere und speichert den Rest.
Verschieben	MOVE	Kopiert Daten von einer Speicheradresse zu einer anderen.
Multiplikation	MUL	Multipliziert zwei Zahlen miteinander.
Negierung	NEG	Negiert (invertiert) die Eingangswerte.
Quadratwurzel	SQRT	Berechnet die Quadratwurzel aus einer Zahl.
Subtraktion	SUB	Subtrahiert eine Zahl von einer anderen.

Verschiedenes

Typ	Abk.	Beschreibung
Alle Programme abbrechen	ABTAL	Bricht alle Programme in den Runtime-Subsystemen ab.
Meldung anzeigen	MSGB	Zeigt eine Meldung in einem Windows-Meldungsfeld an.
Meldungsfenster	MSGW	Zeigt eine Meldung im Programmeditor-Ausgabefenster an.
Bedienerschnittstellenmenü wechseln	Change MMIScreen	Zeigt das angegebene Bedienerschnittstellenmenü an.

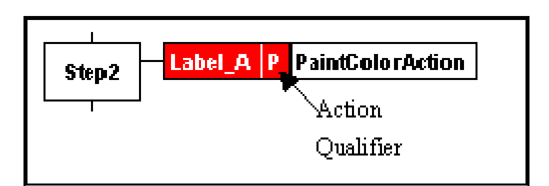
Trigonometrische und logarithmische Funktionen

Typ	Abk.	Beschreibung
Arkuskosinus	ACOS	Berechnet den Arkuskosinus einer Zahl.
Arkussinus	ASIN	Berechnet den Arkussinus einer Zahl.
Arkustangens	ATAN	Berechnet den Arkustangens einer Zahl.
Kosinus	COS	Berechnet den Kosinus einer Zahl.
Exponentialwert	EXP	Berechnet den natürlichen Exponentialwert einer Zahl.
Natürlicher Logarithmus	LN	Berechnet den natürlichen Logarithmus einer Zahl.
Logarithmus	LOG	Berechnet den Logarithmus (Basis 10) einer Zahl.
Sinus	SIN	Berechnet den Sinus einer Zahl.
Tangens	TAN	Berechnet den Tangens einer Zahl.

Ablaufsprachen-Befehlssatz - Zusammenfassung

Anweisung	Beschreibung
Schritt	Ein Schritt stellt einen Zustand dar, in dem das Prozeßverhalten von den mit dem Schritt verknüpften Aktionen definiert wird.
Kontaktplantransition	Erreicht der Stromfluß im Kontaktplan-Strompfad die Ausgangsspule und schaltet sie ein, dann wird die Transition TRUE und läßt den Programmfluß zum nächsten Schritt weitergehen.
Boolesche Transition	Wird der Boolesche Ausdruck TRUE, dann wird die Transition TRUE und läßt den Programmfluß zum nächsten Schritt weitergehen.
Makroschritt	Mit einem Makroschritt können Sie eine Ablaufkette zur Bearbeitung aus einem Schritt einer anderen Ablaufkette heraus aufrufen.
Aktion	Eine Aktion besteht aus Codezeilen, die bearbeitet werden, wenn ein Schritt aktiv wird. Aktions-Bestimmungszeichen geben Einschränkungen bei der Bearbeitung des in der Aktion enthaltenen Kontaktplancode an.
Sprung/Marken-Funktion	Überträgt den Programmfluß zur angegebenen Marke.
Schleife	Führt die AS-Programmausführung wieder zurück zu einer vorangehenden Stelle im Programm und wiederholt so eine Reihe von Schritten.
Auswahlverzweigung	Bei der Auswahlverzweigung kann einer von mehreren Steuerpfaden als der aktive Pfad ausgewählt werden.
Simultanverzweigung	Mit einer Simultanverzweigung können Sie gleichzeitig mehrere Steuerpfad aktivieren.
Programmsteuerungsblock (PRGCB)	Über den Programmsteuerungsblock (PRGCB) kann ein AS-Anwenderprogramm andere KOP- und AS-Anwenderprogramme kompilieren und deren Ausführung steuern.

Bestimmungszeichen erscheinen innerhalb die Aktion wie folgt:



Aktions-Bestimmungszeichen

Wünschen Sie, daß nach aktiv werden des Schritts ...	**... dann verwenden Sie dieses Bestimmungszeichen:**
der Kontaktplan bearbeitet wird und die Bearbeitung stoppt, wenn der Schritt inaktiv wird.	nicht gespeichert (N)
der Kontaktplan bearbeitet wird, bis er durch das Rücksetz-Bestimmungszeichen rückgesetzt wird.	gespeichert (S)
der Kontaktplan einmal bearbeitet wird.	Impuls (P)
der Kontaktplan nach einer Verzögerung* bearbeitet wird. die Kontaktplanbearbeitung stoppt, wenn der Schritt inaktiv wird.	zeitverzögert (D)
die Kontaktplanbearbeitung läuft und stoppt, wenn das Zeitlimit* abläuft oder der Schritt inaktiv wird.	zeitbegrenzt (L)
die Kontaktplanbearbeitung nach einer Verzögerungszeit* begonnen wird und solange läuft, bis sie durch das Rücksetz-Bestimmungszeichen rückgesetzt wird. Setzt ein anderes Aktions-Bestimmungszeichen die Kontaktplanbearbeitung während der Zeitverzögerung zurück, hat dieses Rücksetzen keine Auswirkung, da der Kontaktplan noch nicht gespeichert war. Wird der Schritt inaktiv, ehe die Verzögerungszeit abgelaufen ist, wird der Kontaktplan nicht gespeichert und nie ausgeführt.	verzögert und gespeichert (DS)
die Kontaktplanbearbeitung nach einer Verzögerungszeit* begonnen wird und solange läuft, bis sie durch das Rücksetz-Bestimmungszeichen rückgesetzt wird. Wird eine Aktion während einer Verzögerung rückgesetzt, dann wird der Kontaktplan nicht bearbeitet.	gespeichert und zeitverzögert (SD)
die Kontaktplanbearbeitung begonnen wird und nach der angegebenen Zeit* stoppt. Zum Rücksetzen des Kontaktplans wird ein Rücksetz-Bestimmungszeichen benötigt. Ohne Rücksetzen würde der Kontaktplan nicht wieder bearbeitet. Wird der Schritt inaktiv, wird die Kontaktplanbearbeitung so lange fortgesetzt, bis die Zeit abgelaufen ist. Es wird zwar kein Rücksetz-Bestimmungszeichen benötigt, zum Anhalten der Kontaktplanbearbeitung kann aber einer eingesetzt werden.	gespeichert und zeitbegrenzt (SL)
die Kontaktplanbearbeitung begonnen wird und nach Ablauf der angegebenen Zeit* stoppt. Wird der Schritt inaktiv, wird die Kontaktplanbearbeitung so lange fortgesetzt, bis die Zeit abgelaufen ist. Es wird zwar kein	Impulsbreite (PW)

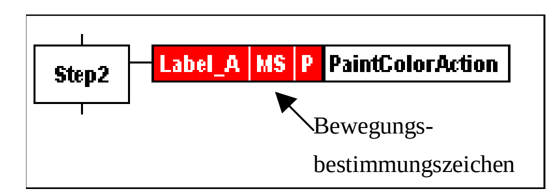
Wünschen Sie, daß nach aktiv werden des Schritts ...	... dann verwenden Sie dieses Bestimmungszeichen:
Rücksetz-Bestimmungszeichen benötigt, zum Anhalten der Kontaktplanbearbeitung kann aber einer eingesetzt werden.	
Der vom Bestimmungszeichen "gespeichert" (S) gestartete Kontaktplan wird vom Rücksetz-Bestimmungszeichen (R) beendet. Sie können das Rücksetz-Bestimmungszeichen verwenden in einer anderen mit dem gleichen Schritt verknüpften Aktion oder in einer Aktion, die mit einem anderen Schritt verknüpft ist. Der Kontaktplan wird zwischen den Schritten bearbeiten. Ist die Aktion mit einem anderen Schritt verknüpft, dann muß die Aktion den gleichen Namen besitzen wie die Aktion, die rückgesetzt werden soll.	Rücksetzen (R)

*Einzelheiten finden Sie unter "Zeitdauer” in Kapitel 4.

Bewegungs-Bestimmungszeichens

Hinweis: Nicht alle Bewegungs-Bestimmungszeichen werden von allen Bewegungsmodulen unterstützt. In der Dokumentation zum Bewegungsmodul finden Sie Hinweise auf die unterstützten Bewegungs-Bestimmungszeichen.

Bewegungs-Bestimmungszeichen geben Bewegungsbedingungen an, die erfüllt sein müssen, damit der Kontaktplan ablaufen kann. Bei Verwendung einer Programmmarke, gilt die Bewegungsbedingung für den Bewegungssatz, der auf die Programmmarke folgt. Bestimmungszeichen erscheinen innerhalb die Aktion wie folgt:



Wählen Sie eines der nachstehenden Bestimmungszeichen aus. Lassen Sie ein Leerzeichen, wenn Sie kein Bestimmungszeichen auswählen.

Soll der Kontaktplan erst bearbeitet werden, wenn ...	... dann verwenden Sie dieses Bestimmungszeichen:
die Bewegung beginnt	Bewegung gestartet (MS)
das Beschleunigungsprofil beendet ist	Beschleunigung beendet (AC)
die Bewegung die Sollgeschwindigkeit erreicht hat	Geschwindigkeit erreicht (AS)
die Bewegung mit dem Bremsprofil beginnt	Verzögerung gestartet (DS)
die Bewegung beendet ist	Bewegung beendet (MC)
die Bewegung beendet ist und alle mit der Bewegung verknüpften Achsen sich innerhalb der Lagetoleranz der programmierten Endpunktes befinden.	In Position (IP)
der Verfahrbefehl beendet ist.	Satzende (EB)

Strukturierter Text Befehlssatz - Zusammenfassung

Anweisungen

Anweisung	Beschreibung
Zuweisung	Ersetzt den Wert eines Objektes mit dem Ergebnis der Auswertung eines Ausdrucks.
CASE	Führt eine Reihe von Anweisungen auf der Grundlage eines Variablenwerts aus.
Kommentar	Zum Einfügen nützlicher Bemerkungen in Ihren Programmcode.
EXIT	Abschluß eines iterativen Prozesses, wie zum Beispiel einer FOR- oder WHILE-Anweisung.
FOR	Verwendet zur wiederholten Ausführung einer Reihe von Anweisungen, wobei die Anzahl Wiederholungen vom Wert einer Laufvariablen abhängt.
Funktionsaufruf	Ausführung eines vordefinierten Algorithmus, mit dem Sie arithmetische oder logische Funktionen, Bitverschiebungen, usw. durchführen können.

Bitfolge

Anweisung	Beschreibung
Anweisung	Beschreibung
AND	Gibt die Boolesche oder bitweise logische UND-Verknüpfung der Eingangswerte zurück.
NOT	Gibt die Boolesche oder bitweise logische Negierung des Eingangswertes zurück.
Exklusiv-ODER (XOR)	Gibt die Boolesche oder bitweise logische Antivalenz der Eingangswerte zurück.
Nach links rotieren (ROL)	Der Rückgabewert ergibt sich aus der kreisförmigen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach links. Die aus der höchstwertigen Stelle (MSB) herausgeschobenen Bits werden zur niedrigstwertigen Bitposition (LSB) rotiert.
Nach rechts rotieren (ROR	Der Rückgabewert ergibt sich aus der kreisförmigen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach rechts. Die aus der niedrigstwertigen Stelle (LSB) herausgeschobenen Bits werden zur höchstwertigen Bitposition (MSB) rotiert.

Nach links verschieben (SHL)	Der Rückgabewert ergibt sich aus der linearen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach links. Die aus der höchstwertigen Bitposition herausgeschobenen Bits gehen verloren, die niedrigstwertigen Bitpositionen werden mit Nullen aufgefüllt.
Nach rechts verschieben (SHR)	Der Rückgabewert ergibt sich aus der linearen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach rechts. Die aus der niedrigstwertigen Bitposition herausgeschobenen Bits gehen verloren, die höchstwertigen Bitpositionen werden mit Nullen aufgefüllt.

Zeichenfolge

Anweisung	Beschreibung
ARRAY_TO_STRING	Wandelt ein Eingangs-Bytefeld um in ASCII-Zeichen und speichert sie in einer Ausgangs-Zeichenfolge.
Ketten	Gibt als Ergebnis zwei verkettete Zeichenfolgen zurück (hängt eine Zeichenfolge an das Ende einer anderen).
Löschen	Gibt als Ergebnis die Eingangszeichenfolge zurück, aus deren Mitte ab einer angegebenen Position die angegebenen Anzahl Zeichen gelöscht wurde.
DSPMSG	Zeigt eine Meldung in einem Fenster im Bedienermenü an.
Gleich (EQ)	Gibt ein Boolesches TRUE zurück, wenn die Eingangswerte gleich sind. Andernfalls wird FALSE zurückgegeben. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Suchen	Gibt die Anfangsposition einer Zeichenfolge innerhalb einer anderen zurück. Wird die Zeichenfolge nicht gefunden, gibt die Anweisung 0 zurück.
Größer als oder gleich (GE)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert größer als der zweite oder gleich ihm ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Größer als (GT)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert größer als der zweite ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
INT_TO_STRING	Konvertiert einen ganzzahligen Wert in eine ASCII-Zeichenfolge.
Einfügen	Gibt eine Zeichenfolge zurück, die entsteht, wenn eine Zeichenfolge an einer angegeben Stelle in eine andere eingefügt wird.
Links	Gibt ab der Position ganz links eine angegebene Anzahl Zeichen der Eingangszeichenfolge zurück.
Länge	Gibt die Länge einer Zeichenfolge zurück.

Kleiner als oder gleich (LE)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert kleiner als der zweite oder gleich ihm ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Kleiner als (LT)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert kleiner als der zweite ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Mitte	Gibt als Ergebnis eine angegebene Anzahl Zeichen ab der angegebenen Position in der Mitte der Eingangs-Zeichenfolge zurück.
MSGWND	Zeigt eine Meldung im Ausgabefenster an.
Ungleich (NE)	Gibt ein Boolesches TRUE zurück, wenn die Eingangswerte ungleich sind. Andernfalls wird FALSE zurückgegeben. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
REAL_TO_STRING	Konvertiert einen reellen Wert in eine ASCII-Zeichenfolge.
Ersetzen	Gibt eine Zeichenfolge zurück, die entsteht, wenn ab einer vorgegebenen Position Zeichen in einer Zeichenfolge durch eine vorgegebene Anzahl Zeichen aus einer anderen Zeichenfolge ersetzt werden.
Rechts	Gibt ab der Position ganz rechts eine angegebene Anzahl Zeichen der Eingangszeichenfolge zurück.

Umwandlung

Anweisung	Beschreibung
BCD_TO_INT	Wandelt einen eingegebenen binär codierten Dezimalwert (BCD) in eine ganze Zahl um.
Bytefeld-in-Zeichenfolge (BATOS)	Wandelt ein Eingangs-Bytefeld um in ASCII-Zeichen und speichert sie in einer Ausgangs-Zeichenfolge. Das abschließende Byte muß Null enthalten.
Datum in Zeichenfolge (DateToString)	Konvertiert ein Datum in eine Zeichenfolge. Eine Pointervariable zeigt auf die Zeichenfolge.
Ganze Zahl in Zeichenfolge (ITOA)	Wandelt eine ganze Zahl in eine ASCII-Zeichenfolge um und gibt das Ergebnis zurück.
REAL in Zeichenfolge (RTOA)	Wandelt eine reelle Zahl in eine ASCII-Zeichenfolge um und gibt das Ergebnis zurück.
RGB in DWORD	Konvertiert ein Tripel aus Rot-, Grün- und Blauwerten in ein DWORD. Wird verwendet bei der ActiveX-Bedienelementfunktion der Bedienerschnittstelle.
STRING_TO_ARRAY	Nimmt die Eingangs-ASCII-Zeichenfolge und speichert die Zeichen der Zeichenfolge in ein Bytefeld.
Zeichenfolge-in-Bytefeld (STOBA)	Wandelt die einzelnen Zeichen der Eingangs-Zeichenfolge in den entsprechenden Dezimalwert um und speichert das Ergebnis in einem Bytefeld. Im letzten Byte des Feldes wird eine Null eingetragen.

Zeichenfolge in Datum (StringToDate)	Wandelt eine Zeichenfolge in ein Datum um und gibt das Ergebnis zurück.
Zeichenfolge in ganze Zahl (ATOI)	Konvertiert eine numerische ASCII-Zeichenfolge in einen ganzzahligen Wert.
Zeichenfolge in REAL (ATOR)	Konvertiert eine numerische ASCII-Zeichenfolge (einschließlich Dezimalpunkt) in ihr reelles Äquivalent.
Zeichenfolge in Uhrzeit (StringToTOD)	Wandelt eine Zeichenfolge in eine Uhrzeit um und gibt das Ergebnis zurück.
Uhrzeit in Zeichenfolge (TODToString)	Wandelt eine Uhrzeit in eine Zeichenfolge um und legt das Ergebnis in den durch pString angezeigten Symbol ab.

Zähler und Timer

Anweisung	Beschreibung
Abwärts zählen (CTD)	Zählt von einem Vorgabewert zurück auf Null. Der Ausgang (Q) wird TRUE, wenn der Zählwert Null ist. Kann zum Beispiel zum Zählen sich wiederholender Ereignisse benutzt werden.
Aufwärts zählen (CTU)	Zählt von Null hoch zu einem Vorgabewert. Der Ausgang (Q) wird TRUE, wenn der Zählwert gleich dem Vorgabewert ist. Kann zum Beispiel zum Zählen sich wiederholender Ereignisse benutzt werden.
Aufwärts/abwärts zählen (CTUD)	Zählt nach oben oder unten; setzt einen Aufwärtszählerausgang, wenn der aktuelle Zählwert größer oder gleich zum Vorgabewert ist, oder einen Abwärtszählerausgang, wenn der aktuelle Zählwert kleiner oder gleich Null ist.
Abschaltverzögerung (TOF)	Schaltet einen Ausgang nach einer voreingestellten Verzögerungszeit ab.
Einschaltverzögerung (TON)	Schaltet einen Ausgang nach einer voreingestellten Verzögerungszeit ein.
Timerimpuls (TP)	Der Funktionsblock TP mißt die Dauer eines Ereignisses. Nachdem sein Eingang von AUS nach EIN gepulst ist, behält der TP die Zeit zum voreingestellten Intervall und setzt den Ausgang auf FALSE. Hierdurch wird der TP zur Ausschaltverzögerung.

Flankenerkennung

Anweisung	Beschreibung
Triggersignal bei abfallender Flanke (F_TRIG)	Der Funktionsblock F_TRIG setzt einen Ausgang einen Zyklus lang auf TRUE, wenn der Eingang des Funktionsblock von TRUE auf FALSE wechselt.
Triggersignal bei steigender Flanke (R_TRIG)	Der Funktionsblock R_TRIG setzt einen Ausgang einen Zyklus lang auf TRUE, wenn der Eingang des Funktionsblock von FALSE auf TRUE wechselt.

Erweiterte Timer

Anweisung	Beschreibung
Erweiterte Timer-Ausschaltverzögerung (XTOF)	Schaltet einen Ausgang nach einer voreingestellten Verzögerungszeit ab. Wie der TOF-Timer, jedoch mit zwei zusätzlichen Eingängen (LoadTime und NewTime).
Erweiterte Timer-Einschaltverzögerung (XTON)	Schaltet einen Ausgang nach einer voreingestellten Verzögerungszeit ein. Wie der TON-Timer, jedoch mit zwei zusätzlichen Eingängen (LoadTime und NewTime).
Erweiterter Timer-Impuls (XTP)	Der Funktionsblock XTP mißt die Dauer eines Ereignisses. Nachdem sein Eingang von AUS nach EIN gepulst ist, behält der XTP die Zeit zum voreingestellten Intervall und setzt den Ausgang auf FALSE. Hierdurch wird der XTP zur Ausschaltverzögerung. Wie der TP-Timer, jedoch mit zwei zusätzlichen Eingängen (LoadTime und NewTime).

Dateifunktionen

Anweisung	Beschreibung
APPENDFILE	Schreibt Daten an das Ende der durch den Dateisteuerungsblock spezifizierten Datei.
BREAK	Stoppt das Programm, wenn Sie Austesten freigegeben haben.
CLOSEFILE	Schließt eine Datei, die von der Funktion OPENFILE geöffnet wurde.
COPYFILE	Kopiert eine bestehende Datei.
DELETEFILE	Löscht eine bestehende Datei.
NEWFILE	Erstellt eine neue Datei.
OPENFIE	Öffnet eine Datei für eine Operation (z.B. lesen oder schreiben).
READFILE	Liest Daten aus einer Datei.
REWINDFILE	Setzt den internen Dateipointer auf den Anfang einer Datei.
WRITEFILE	Schreibt Daten in eine Datei.

Verschiebe-/Rotationsfunktionen

Anweisung	Beschreibung
ROL	Rotiert die einzelnen Bits eines Wertes um eine angegebene Anzahl Stellen nach links.
ROR	Rotiert die einzelnen Bits eines Wertes um eine angegebene Anzahl Stellen nach rechts.
SHL	Verschiebt die einzelnen Bits eines Wertes um eine angegebene Anzahl Stellen nach links.

Anweisung	Beschreibung
SHR	Verschiebt die einzelnen Bits eines Wertes um eine angegebene Anzahl Stellen nach rechts.
IF	Führt eine Reihe von Anweisungen nur aus, wenn eine Boolesche Variable TRUE ist.
INCLUDE	Ruft eine externe Datei auf und führt die in der Datei enthaltene Reihe von Anweisungen aus.
REPEAT	Führt eine Reihe von Anweisungen wiederholt solange aus, bis eine Boolesche Variable TRUE wird.
SCAN	Hält die Ausführung der strukturierten Textanweisungen an, während ein E/A-Zyklus ausgeführt wird.
WHILE	Führt eine Reihe von Anweisungen wiederholt solange aus, bis eine Boolesche Variable FALSE wird.

Arithmetische Funktionen

Anweisung	Beschreibung
Addition (ADD)	Gibt als Ergebnis die Summe der Eingangswerte zurück.
ABS	Berechnet den Absolutwert des Eingangswertes.
Division (DIV)	Gibt das Ergebnis der Division des ersten Eingangswertes durch den zweiten Eingangswert zurück.
Exponent (EXPT)	Gibt das Ergebnis der Potenzierung des ersten Wertes durch den zweiten Wert zurück.
MAX	Bestimmt den größeren von zwei Werten.
MIN	Bestimmt den kleineren von zwei Werten.
Modulo (MOD)	Gibt den Rest aus dem Ergebnis der Division des ersten Eingangswertes durch den zweiten Eingangswert zurück.
Verschieben (MOVE)	Gibt das Ergebnis der Konvertierung des Eingangswertes in den Datentyp des Ausgangs zurück.
Multiplikation (MUL)	Gibt das Produkt der Multiplikation der Eingangswerte zurück.
Negation (NEG)	Gibt das Ergebnis eines Vorzeichenwechsels beim Eingangssignal zurück.
Quadratwurzel (SQRT)	Gibt die Quadratwurzel des Eingangswertes zurück.
Subtraktion (SUB)	Gibt das Ergebnis einer Subtraktion des zweiten Eingangswertes vom ersten Eingangswert zurück.

Verschiedenes

Anweisung	Beschreibung
Alle Abbrechen	Bricht **alle** Programme in den Runtime-Subsystemen ab.
Meldung anzeigen	Zeigt eine Meldung in einem Windows-Meldungsfeld an. Die Programmausführung wird ohne Unterbrechung fortgesetzt, während die Meldung angezeigt wird. Der Bediener kann auf *OK* klicken, um die Meldung auszublenden.
Feld initialisieren	Setzt alle Elemente eines Feld auf einen angegebenen Anfangswert.
Meldungsfenster	Zeigt eine Meldung im Programmeditor-*Ausgabefenster* an.
Bedienerschnittstellen menü wechseln	Zeigt das angegebene Bedienerschnittstellenmenü an.

Auswahl

Anweisung	Beschreibung
Maximum (MAX)	Bestimmt den größten Wert an den Eingängen und gibt diesen Wert als Ausgangswert zurück. Die Funktion unterstützt bis zu 16 Eingangswerte.
Minimum (MIN)	Bestimmt den kleinsten Wert an den Eingängen und benutzt diesen Wert als Ausgangswert. Die Funktion erlaubt bis zu 16 Eingangswerte.

Systemobjekte

Anweisung	Beschreibung
PID-Regelung (PID)	PID ist eine Anweisung, die die automatische Regelung kontinuierlicher Prozesse erlaubt. Für jeden Kreis führt die Anweisung Proportionalregelung und wahlweise Integralregelung, Differentialregelung, oder beide durch:
Programmsteuerungsbl ock (PRGCB)	Über den Programmsteuerungsblock (PRGCB) kann ein AS-Anwenderprogramm andere KOP- und AS-Anwenderprogramme kompilieren und deren Ausführung steuern.
Timer (TMR)	Realisiert einen Timer in strukturierter Textsprache.

Trigonometrische und logarithmische Funktionen

Anweisung	Beschreibung
Arkuskosinus (ACOS)	Berechnet den Arkuskosinus des Eingangswertes. Das Ergebnis ist im Bogenmaß.
Arkussinus (ASIN)	Berechnet den Arkussinus des Eingangswertes. Das Ergebnis ist im Bogenmaß.
Arkustangens (ATAN)	Berechnet den Arkustangens des Eingangswertes. Das Ergebnis ist im Bogenmaß.
Kosinus (COS)	Berechnet den Kosinus des Eingangswerts, der im Bogenmaß angegeben werden muß. .
Exponentialwert (EXP)	Berechnet den natürlichen Exponentialwert des Eingangswertes (erhebt *e* zur Potenz des Eingangswertes).
EXPT	Erhebt einen Wert zur Potenz des zweiten Wertes.
Natürlicher Logarithmus (LN)	Berechnet den natürlichen Logarithmus eines Wertes.
Logarithmus (LOG)	Berechnet den dekadischen Logarithmus eines Wertes.
Sinus (SIN)	Berechnet den Sinus des Eingangswerts, der im Bogenmaß angegeben werden muß.
Tangens (TAN)	Berechnet den Tangens des Eingangswerts, der im Bogenmaß angegeben werden muß.

Bewegungsfunktionen

AXSJOG	Startet einen Tippbefehl an einer angegebenen Achse.
MOVEAXS	Startet einen Verfahrbefehl an einer angegebenen Achse.
STOPJOG	Stoppt einen Tippbefehl an einer angegebenen Achse.

Anweisungslisten-Befehlssatz - Zusammenfassung

Bitfolge

Anweisung	Beschreibung
AND	Gibt die Boolesche oder bitweise logische UND-Verknüpfung der Eingangswerte zurück.
NOT	Gibt die Boolesche oder bitweise logische Negierung des Eingangswertes zurück.
Exklusiv-ODER (XOR)	Gibt die Boolesche oder bitweise logische Antivalenz der Eingangswerte zurück.
Nach links rotieren (ROL)	Der Rückgabewert ergibt sich aus der kreisförmigen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach links. Die aus der höchstwertigen Stelle (MSB) herausgeschobenen Bits werden zur niedrigstwertigen Bitposition (LSB) rotiert.
Nach rechts rotieren (ROR)	Der Rückgabewert ergibt sich aus der kreisförmigen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach rechts. Die aus der niedrigstwertigen Stelle (LSB) herausgeschobenen Bits werden zur höchstwertigen Bitposition (MSB) rotiert.
Nach links verschieben (SHL)	Der Rückgabewert ergibt sich aus der linearen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach links. Die aus der höchstwertigen Bitposition herausgeschobenen Bits gehen verloren, die niedrigstwertigen Bitpositionen werden mit Nullen aufgefüllt.
Nach rechts verschieben (SHR)	Der Rückgabewert ergibt sich aus der linearen Verschiebung der Bits des Eingangswertes um die angegebenen Anzahl Stellen nach rechts. Die aus der niedrigstwertigen Bitposition herausgeschobenen Bits gehen verloren, die höchstwertigen Bitpositionen werden mit Nullen aufgefüllt.

Zeichenfolge

Anweisung	Beschreibung
Ketten	Gibt als Ergebnis zwei verkettete Zeichenfolgen zurück (hängt eine Zeichenfolge an das Ende einer anderen).
DELETE	Gibt als Ergebnis die Eingangszeichenfolge zurück, aus deren Mitte ab einer angegebenen Position die angegebenen Anzahl Zeichen gelöscht wurde.
FIND	Gibt die Anfangsposition einer Zeichenfolge innerhalb einer anderen zurück. Wird die Zeichenfolge nicht gefunden, gibt die Anweisung 0 zurück.

Anweisung	Beschreibung
INSERT	Gibt eine Zeichenfolge zurück, die entsteht, wenn eine Zeichenfolge an einer angegeben Stelle in eine andere eingefügt wird.
LEFT	Gibt ab der Position ganz links eine angegebene Anzahl Zeichen der Eingangszeichenfolge zurück.
Länge	Gibt die Länge einer Zeichenfolge zurück.
Mitte	Gibt als Ergebnis eine angegebene Anzahl Zeichen ab der angegebenen Position in der Mitte der Eingangs-Zeichenfolge zurück.
REPLACE	Gibt eine Zeichenfolge zurück, die entsteht, wenn ab einer vorgegebenen Position Zeichen in einer Zeichenfolge durch eine vorgegebene Anzahl Zeichen aus einer anderen Zeichenfolge ersetzt werden.
Rechts	Gibt ab der Position ganz rechts eine angegebene Anzahl Zeichen der Eingangszeichenfolge zurück.
Gleich (EQ)	Gibt ein Boolesches TRUE zurück, wenn die Eingangswerte gleich sind. Andernfalls wird FALSE zurückgegeben. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Größer als oder gleich (GE)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert größer als der zweite oder gleich ihm ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Größer als (GT)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert größer als der zweite ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Kleiner als oder gleich (LE)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert kleiner als der zweite oder gleich ihm ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Kleiner als (LT)	Gibt ein Boolesches TRUE zurück, wenn der erste Eingangswert kleiner als der zweite ist. Andernfalls kommt FALSE zurück. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.
Ungleich (NE)	Gibt ein Boolesches TRUE zurück, wenn die Eingangswerte ungleich sind. Andernfalls wird FALSE zurückgegeben. Setzt den Ausgangs des Kontaktplan-Strompfads entsprechend.

Umwandlung

Anweisung	Beschreibung
Bytefeld-in-Zeichenfolge (BATOS)	Wandelt ein Eingangs-Bytefeld um in ASCII-Zeichen und speichert sie in einer Ausgangs-Zeichenfolge. Das abschließende Byte muß Null enthalten.
Datum in Zeichenfolge (DateToString)	Konvertiert ein Datum in eine Zeichenfolge. Eine Pointervariable zeigt auf die Zeichenfolge.

Anweisung	Beschreibung
Ganze Zahl in Zeichenfolge (ITOA)	Wandelt eine ganze Zahl in eine ASCII-Zeichenfolge um und gibt das Ergebnis zurück.
REAL in Zeichenfolge (RTOA)	Wandelt eine reelle Zahl in eine ASCII-Zeichenfolge um und gibt das Ergebnis zurück.
RGB in DWORD	Konvertiert ein Tripel aus Rot-, Grün- und Blauwerten in ein DWORD. Wird verwendet bei der ActiveX-Bedienelementfunktion der Bedienerschnittstelle.
Zeichenfolge-in-Bytefeld (STOBA)	Wandelt die einzelnen Zeichen der Eingangs-Zeichenfolge in den entsprechenden Dezimalwert um und speichert das Ergebnis in einem Bytefeld. Im letzten Byte des Feldes wird Null eingetragen.
Zeichenfolge in Datum (StringToDate)	Wandelt eine Zeichenfolge in ein Datum um und gibt das Ergebnis zurück.
Zeichenfolge in ganze Zahl (ATOI)	Konvertiert eine numerische ASCII-Zeichenfolge in einen ganzzahligen Wert. Nicht numerische ASCII-Zeichen oder numerische Zeichen, die auf ein nicht numerisches Zeichen folgen,werden ignoriert.
Zeichenfolge in REAL (ATOR)	Konvertiert eine numerische ASCII-Zeichenfolge (einschließlich Dezimalpunkt) in ihr reelles Äquivalent. Nicht numerische ASCII-Zeichen oder numerische Zeichen, die auf ein nicht numerisches Zeichen folgen,werden ignoriert.
Zeichenfolge in Uhrzeit (StringToTOD)	Wandelt eine Zeichenfolge in eine Uhrzeit um und gibt das Ergebnis zurück.
Uhrzeit in Zeichenfolge (TODToString)	Wandelt eine Uhrzeit in eine Zeichenfolge um und legt das Ergebnis in den durch pString angezeigten Symbol ab.

Arithmetische Funktionen

Anweisung	Beschreibung
Absolutwert (ABS)	Gibt den Absolutwert des Eingangswertes zurück.
Addition (ADD)	Gibt als Ergebnis die Summe der Eingangswerte zurück.
Division (DIV)	Gibt das Ergebnis der Division des ersten Eingangswertes durch den zweiten Eingangswert zurück.
Exponent (EXPT)	Gibt das Ergebnis der Potenzierung des ersten Wertes durch den zweiten Wert zurück.
Modulo (MOD)	Gibt den Rest aus dem Ergebnis der Division des ersten Eingangswertes durch den zweiten Eingangswert zurück.
Verschieben (MOVE)	Gibt das Ergebnis der Konvertierung des Eingangswertes in den Datentyp des Ausgangs zurück.
Multiplikation (MUL)	Gibt das Produkt der Multiplikation der Eingangswerte zurück.
Negation (NEG)	Gibt das Ergebnis eines Vorzeichenwechsels beim Eingangssignal zurück.
Quadratwurzel	Gibt die Quadratwurzel des Eingangswertes zurück.

(SQRT)	
Subtraktion (SUB)	Gibt das Ergebnis einer Subtraktion des zweiten Eingangswertes vom ersten Eingangswert zurück.

Verschiedenes

Anweisung	Beschreibung
Alle abbrechen	Bricht **alle** Programme in den Runtime-Subsystemen ab.
Bedienerschnittstellen menü wechseln	Zeigt das angegebene Bedienerschnittstellenmenü an.

Trigonometrische und logarithmische Funktionen

Anweisung	Beschreibung
Arkuskosinus (ACOS)	Berechnet den Arkuskosinus des Eingangswertes. Das Ergebnis ist im Bogenmaß.
Arkussinus (ASIN)	Berechnet den Arkussinus des Eingangswertes. Das Ergebnis ist im Bogenmaß.
Arkustangens (ATAN)	Berechnet den Arkustangens des Eingangswertes. Das Ergebnis ist im Bogenmaß.
Kosinus (COS)	Berechnet den Kosinus des Eingangswerts, der im Bogenmaß angegeben werden muß. **.**
Exponentialwert (EXP)	Berechnet den natürlichen Exponentialwert des Eingangswertes (erhebt *e* zur Potenz des Eingangswertes).
Natürlicher Logarithmus (LN)	Berechnet den natürlichen Logarithmus eines Wertes.
Logarithmus (LOG)	Berechnet den dekadischen Logarithmus eines Wertes.
Sinus (SIN)	Berechnet den Sinus des Eingangswerts, der im Bogenmaß angegeben werden muß.
Tangens (TAN)	Berechnet den Tangens des Eingangswerts, der im Bogenmaß angegeben werden muß.

Anhang
H

Glossar der Fachausdrücke

.CFG

Eine Datei mit der Dateinnamenerweiterung CFG enthält Informationen über die Hardware-Konfigurationsparameter des Systems und symbolische Information, die sich auf physikalische Komponenten und die E/A-Signale bezieht.

Aktions-Bestimmungszeichen

Graphisches Programmierelement in einer Ablaufkette, das mit einem Aktionsblock verknüpft ist und die Ausführung der Aktionslogik bezüglich des Zeitraums steuert, in dem der zugehörige Schritt aktiv ist.

Aktion

Mit einem Namen versehene Sammlung von Operationen, die mit einem oder mehreren Schritten in einer Ablaufkette verknüpft ist.

Aktive Datei

Die Programmdatei im obersten Programmeditor-Fenster, deren Kopfleiste hervorgehoben ist. Befehle, die aus den Menüs heraus oder durch Klicken auf Schaltflächen in Symbolleisten ausgeführt werden, wirken sich immer auf die aktive Datei aus.

ANY

Generischer Datentyp, der für einen beliebigen unterstützten Datentyp stehen kann.

ANY_BIT

Generischer Datentyp, der für diese Datentypen stehen kann: DWORD, WORD, BYTE, BOOL (einschließlich eines einzelnen Bits innerhalb dieser Datentypen).

ANY_INT

Generischer Datentyp, der für den Datentyp INT, UINT oder DINT stehen kann.

ANY_NUM

Generischer Datentyp, der für diese Datentypen stehen kann: ANY_REAL und ANY_INT.

ANY_REAL

Generische Typenbezeichnung, die für den Datentyp REAL stehen kann.

Anwendungssymbol

Ein Anwendungssymbol ist eine Schritt-Mustervorlage mit angehängtem Symbol. Wird ein Anwendungssymbol in einer Ablaufkette plaziert, dann wird ein neuer Schritt angelegt und die Informationen aus dem Anwendungssymbol werden in diesen Schritt kopiert.

Platine

Siehe E/A-Karte

BOOL

Element der Datentypgruppe ANY_BIT. BOOL Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_BIT oder BOOL akzeptieren. Ein Boolescher Wert ist ein Bit lang und kann einen von zwei Werten annehmen: TRUE (1 oder EIN) oder FALSE (0 oder AUS).

Boolesche Transition

Transitionstyp in einer Ablaufkette, der durch einen Ausdruck aus Symbolen und Operatoren besteht, die ein einziges Boolesches Ergebnis liefern. Ist das Boolesche Ergebnis TRUE, dann ist die AS-Transitionsbedingung erfüllt.

Boolesche

Logisches Element oder logischer Ausdruck, dessen Ergebnis entweder TRUE oder FALSE ist.

BYTE

Element der Datentypgruppe ANY_BIT. BYTE Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_BIT oder BYTE akzeptieren. Der Datentyp BYTE stellt eine ganze Zahl ohne Vorzeichen dar, die aus einer oder mehreren Ziffern (0-9) besteht und keinen Dezimalpunkt enthalten darf. Ein BYTE ist 8 Bits lang und hat einen Bereich von 0 bis 255.

Karte

Siehe E/A-Karte

Zeichenposition

Position eines Zeichens innerhalb einer Zeichenfolge. Das erste Zeichen einer Zeichenfolge beginnt bei Position 1.

Spule

Graphisches Kontaktplanelement, das ein Boolesches Ausgangssymbol darstellt.

Konfigurationsdatei

Eine Konfigurationsdatei enthält Daten und Parameter, die die Hardwarekonfiguration des Systems beschreiben. Diese Datei enthält auch symbolische Bezeichner, die im Anwenderprogramm zur Adressierung der physikalischen Hardware oder E/A-Signale verwendet werden.

Steckverbinder

Gruppe von E/A-Ports, die normalerweise zu einem physikalischen Steckverbinder auf einem Modul geführt werden.

Kontaktwerkzeug

Eine Schaltfläche auf der Kontaktplan-Symbolleiste, mit der ein Eingangskontakt in einen Kontaktplan eingefügt wird.

Kontakt

Graphisches Kontaktplanelement, das ein Boolesches Eingangssymbol darstellt.

Steuerschleifenelement

Siehe Schleife.

Steuerpfad

Ein Steuerpfad ist eine Linie zwischen Schritten in einer Ablaufkette, die den Pfad des Programmsteuerungsablaufs anzeigt.

Cursor

Der Cursor ist das Objekt, das der Bewegung des Zeigegerätes (Tastatur, Maus, Rollkugel oder Touchscreen) folgt und mit dem die Elemente der Bedienerschnittstelle am Bildschirm ausgewählt werden.

DATE

Element der Datentypgruppe ANY_DATE. DATE Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_DATE oder DATE akzeptieren.

DINT

Element der Datentypgruppe ANY_NUM. DINT Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_NUM, ANY_INT oder DINT akzeptieren. Der Datentyp DINT stellt eine ganze Zahl mit Vorzeichen dar, die aus einer oder mehreren Ziffern (0-9) besteht und keinen Dezimalpunkt enthalten darf. Ein DINT ist 32 Bits lang und besitzt einen Bereich von -2147483648 bis +2147483647.

DWORD

Element der Datentypgruppe ANY_BIT. DWORD Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_BIT oder DWORD akzeptieren. Der Datentyp DWORD stellt eine ganze Zahl ohne Vorzeichen dar, die aus einer oder mehreren Ziffern (0-9) besteht und keinen Dezimalpunkt enthalten darf. Ein DWORD ist 32 Bits lang und hat einen Bereich von 0 bis 4.294.967.295.

Dynamischer Datenaustausch

DDE ist der Übergang von Daten zwischen Anwendungen, der ohne Beteiligung oder Überwachung durch den Anwender vonstatten geht. In einer Windows Umgebung wird DDE durch einen Satz Meldungstypen, empfohlenen Prozeduren (Protokollen) zur Bearbeitung dieser Meldungstypen, sowie einigen neu definierten Datentypen durchgeführt. Wenn diese Protokolle befolgt werden, können

voneinander unabhängig erstellte Anwendungen Daten untereinander austauschen, ohne daß der Anwender dabei mitwirken muß.

FCB

Dateisteuerungsblock. Ein internes Datenelement, mit dem die Datei-E/A-Operationen in strukturiertem Text gesteuert werden.

Erster Zyklus

Mit der Logik des ersten Zyklus werden spezielle Befehle nur einmal während des ersten Zyklus eines Programms ausgeführt. Beispiel: Timer und Zähler werden auf einen Anfangswert gesetzt.

Funktion

Vordefinierte AS-Algorithmen, die eine einzelne Operation (z.B. Quadratwurzel, nach links rotieren, Datei zurückspulen usw.) ausführen.

Funktionsblock

Vordefinierte Kontaktplanalgorithmen, die eine einzelne Operation ausführen, mit der die Möglichkeiten des Programms über die einfache Logik von Kontakten und Spulen hinaus erweitert wird.

Mensch-Maschinen-Schnittstelle

Die Mensch-Maschinen-Schnittstelle ist das Funktionspaket, über das der menschlische Bediener mit der Maschine oder Prozeßsteuerung verkehren kann.

E/A

E/A bezeichnet die Eingänge und Ausgänge der Steuerung.

E/A-Karte

Eine E/A-Karte wird in einen Erweiterungssteckplatz der Systemeinheit gesteckt und bildet die Verbindung zu den dezentralen E/A-Chassis und Modulen. Wird auch Karte oder Modul genannt.

E/A-Zyklussteuerung

Die E/A-Zyklussteuerung ist eine Task in dem System, die verantwortlich ist, alle Eingangssignale von der E/A-Schnittstelle zu erfassen und alle Ausgangssignale

zur E/A zu übertragen. Die E/A-Zyklussteuerung aktualisiert die internen E/A-Datentabellen, die von den anderen Subsystemen verwendet werden.

Symbol

Ein Symbol ist ein Bild, das ein anderes Objekt darstellt. In AS+ kann es zur Darstellung von Schrittboxen oder Makoschritten verwendet werden. Ein Symbol kann vordefinierte Funktionen enthalten und für Funktionsbibliotheken oder Festroutinen verwendet werden.

IEC 1131-3

IEC 1131-3 ist eine internationale Norm von der International Electrotechnical Commission, die die Programmiersprache für speicherprogrammierbare Steuerungen definiert.

Anweisungsliste

Die Anweisungsliste ist eine Textsprache, die in IEC 1131-3 definiert wird.

INT

Element der Datentypgruppe ANY_NUM. INT Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_NUM, ANY_INT oder INT akzeptieren. Der Datentyp INT stellt eine ganze Zahl mit Vorzeichen dar, die aus einer oder mehreren Ziffern (0-9) besteht und keinen Dezimalpunkt enthalten darf. In einer Erweiterung der IEC1131-3 Spezifikation, ist INT 32 Bits lang und hat einen Bereich von -2.147.483.648 bis +2.147.483.647.

Ganze Zahl

Der ganzahlige Datentyp INT besteht aus 32 Bits mit Vorzeichen und kann Werte zwischen -2.147.483.648 und +2.147.483.648 darstellen.

Marke

Graphisches Programmierelement in einem AS- oder KOP-Programm, das festlegt, wo der Programmfluß nach dem zugehörigen Sprung fortgesetzt wird.

Kontaktplanprogramm

Ein Kontaktplanprogramm ist eine in IEC 1131 definierte graphische Programmiersprache mit Kontaktplanlogik.

Schleife

Ein graphisches Programmierelement in einer Ablaufkette, das die Programmausführung entweder gerade nach unten weiterführt oder aber wieder zu einem Punkt weiter oben im Steuerpfad zurückführt. Das Schleifenelement enthält zwei Transitionsbedingungen: Eine Bedingung führt den Programmfluß nach unten weiter, während die andere den Programmfluß wieder nach oben zurücklenkt. Mehrere Schleifenelemente können ineinander verschachtelt werden, dürfen sich aber nicht kreuzen und dürfen nicht in Auswahlverzweigung oder Parallelverzweigungen eintreten.

Schleifenwerkzeug

Das Schleifenwerkzeug ist eine Schaltfläche auf der AS-Bearbeitungs-Symbolleiste, mit der ein Steuerungselement in eine Ablaufkette eingefügt werden kann.

Makroschritt

Mit einem Namen versehenes graphisches Element, das die Einfügung einer vollständigen Ablaufkette als einen einzelnen Schritt darstellt. Die eingefügte Ablaufkette beginnt ihre Ausführung bei ihrem Startschritt, wenn der sie rufende Makroschritt aktiv wird. Der Makroschritt ist abgeschlossen, wenn die eingebundene Ablaufkette den Endeschritt erreicht. Makroschritte können zur Steuerung der Komplexität oder zur Erzielung einer globaleren Ansicht einer aus einer Sammlung von kleinen und einfacheren Ablaufketten bestehenden Ablaufkette verwendet werden, Aktionen können so an Makroschritte angehängt werden, wie sie auch an reguläre Schritte angehängt werden.

Bedienerschnittstelle

Die Bedienerschnittstelle besteht aus einer Reihe von Computermenüs, Meldungen und Fenstern, die dem Bediener zur Überwachung und Steuerung einer Maschine oder eines Prozesses zur Verfügung gestellt werden.

Parallelverzweigung

Graphisches Programmierelement in Ablaufsprache, das einen Steuerpfad in zwei oder mehr Parallelpfade aufteilt. Erreicht die Programmausführung den Anfang einer Parallelverzweigung, dann werden alle nachfolgenden Steuerpfade gleichzeitig aktiv. Diese Steuerpfade sind solange aktiv, bis alle Steuerpfade innerhalb der Parallelverzweigung den Zusammenführungspunkt erreichen. An diesem Punkt werden alle Pfade innerhalb der Parallelverzweigung deaktiviert und der Steuerpfad unterhalb des Zusammenführungspunktes werden aktiv.

Port

Der E/A-Port besteht normalerweise aus acht E/A-Bits. Auf einen Port kann als ganze Zahl innerhalb von Programmen zugegriffen werden.

PRGCB

Mit dem Programmsteuerungsblock können Ablaufketten andere Anwenderprogramme erstellen und deren Ausführung steuern.

Programmeditor

Der Programmeditor ist ein Dienstprogramm,mit dem Anwenderprogramme bearbeitet werden können. Er besteht aus Editoren für AS- und KOP-Programme.

Projekt

Organisiert oder gruppiert die Anwenderprogramme und Konfigurationsdateien für eine Anwendung in einem eigenen Unterverzeichnis.

REAL

Element der Datentypgruppe ANY_NUM. REAL Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_NUM, ANY_REAL oder REAL akzeptieren. Der Datentyp REAL ist ein vorzeichenbehafteter Wert, der aus einer oder mehreren Ziffern (0-9) besteht und einen Dezimalpunkt enthält. Der Bereich der REAL-Zahlen ist: -3,402823 E38 bis -1,401298 E-45 (negativ) bzw. +1,401298E-45 bis +3,402823 E38 (positiv).

Kontaktplanlogik

Kontaktplanlogik ist die graphische Programmiersprache, die zur Beschreibung von Anwenderprogrammlogik auf der Grundlage von Analogien zu elektrischen Relaiskontakten und Spulen verwendet wird.

KOP

KOP steht für Kontaktplanlogik. Eine graphische Programmiersprache auf der Grundlage von Analogien zu elektrischen Relaiskontakten und Spulen.

Kontaktplantransition

Transition in einer Ablaufkette, die mit Kontaktplanlogik programmiert wurde. Die Kontaktplantransition besteht aus einem einzelnen Strompfad mit einer Ausgangsspule, die den gleichen Namen wie die Kontaktplantransitionsobjekt trägt. Erhält diese Ausgangsspule Stromfluß, dann ist die AS-Transitionsbedingung erfüllt.

Strompfad

Graphisches Kontaktplanelement, das eine Kontaktplanfunktion darstellt.

Runtimemaschine

Modul, das für die Laufzeitzuweisung und Ausführung der mit dem Projekt-Quellcode verknüpften Programmlogik (z.B. AS, KOP, usw.) verantwortlich ist. Dieses Modul arbeitet auch als ein Server zu DDE-Clients.

Auswahlverzweigung

Graphisches Programmierelement in Ablaufsprache, das einen einzelnen Steuerpfad in zwei oder mehr Pfade aufteilt. Der zur Bearbeitung ausgewählte Steuerpfad wird von der Transitionsbedingung bestimmt, die am Anfang der einzelnen neuen Steuerpfade steht.

Ablaufsprache

AS ist die graphische Programmiersprache zur Diagrammdarstellung von Ablauflogik mit Schritten, Transitionen und Aktionen.

AS

AS steht für Ablaufsprache, der graphischen Programmiersprache zur Diagrammdarstellung von Ablauflogik mit Schritten, Transitionen und Aktionen. IEC 1131 hat eine spezielle Implementierung der AS zum Gebrauch mit SPS definiert.

AS+

AS+ ist eine erweiterte Version der von PC Control verwendeten Standard-Programmiersprache IEC 1131 Ablaufsprache.

Ablaufketten-Transitionsmerker

Graphisches AS-Programmierelement, das mit einem Schritt oder Makroschritt verknüpft werden kann und zur Programmablaufsteuerung eingesetzt wird. In Verbindung mit einem Schritt kann der Ablaufketten-Transitionsmerker den Schritt abbrechen und den Programmfluß zu einem anderen Schritt lenken. In Verbindung mit einem Makroschritt, kann der Ablaufketten-Transitionsmerker die Tochter-Ablaufkette abbrechen und den Programmfluß zu einem anderen Schritt in der Mutter-Ablaufkette umlenken.

Simultanverzweigung

Simultanverzweigung ist ein graphisches AS-Programmierelement, das einen Steuerpfad in zwei oder mehr parallele Steuerpfade aufteilt. Erreicht die Programmausführung den Anfang einer Simultanverzweigung, dann werden alle nachfolgenden Steuerpfade gleichzeitig aktiv. Diese Steuerpfad sind solange aktiv, bis alle Steuerpfade in der Simultanverzweigung einen Zusammenführungspunkt erreicht haben. An diesem Punkt werden die Pfade deaktiviert und der Steuerpfad unterhalb der Zusammenführung wird aktiv.

Steckplatz

Ein Platz in einer E/A oder einem Computer, in dem eine Karte eingesetzt werden kann.

Schritt

Benanntes graphisches Element in einer Ablaufkette, das einen Zustand oder eine Zeitspanne in der Programmausführung darstellt, während der die mit dem Schritt verknüpften Aktionen und Funktionen ausgeführt werden.

Schrittname

Der Schrittname ist ein Bezeichner, der einen Schritt in einer Ablaufkette anspricht. Sie können den Schrittnamen verändern, indem Sie auf das Schrittkästchen doppelklicken, um die Schrittbearbeitungs-Dialogbox zu öffnen, und dann auf den Bezeichner in der Schrittnamen-Editierbox doppelklicken und einen neuen Bezeichner eingeben.

STRING

Element der Datentypgruppe ANY. STRING Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY oder STRING akzeptieren. Das Format eines STRING-Datentyps besteht aus einer Zeichenfolge mit bis zu 250 ASCII-Zeichen in einzelnen Anführungszeichen.

Strukturierter Text

Strukturierter Text ist eine textstrukturierte Programmiersprache entsprechend IEC 1131.

Symbol

Interne Speicheradresse, die Informationen enthält. Die Informationen selbst werden vom Datentyp bestimmt; es können reelle Zahlen, ganze Zahlen, Zeichenfolgen usw. sein. Mit dem Symbolmanager wird ein Symbol definiert und erhält eine symbolische Bezeichnung und einen Datentyp zugewiesen.

TIME

Element der Datentypgruppe ANY. TIME Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY oder TIME akzeptieren. Das Format des TIME-Datentyps besteht aus einem T# oder t#, dem eine Folge mit einer oder mehreren Zahlen und Zeiteinheit-Kennungen folgt. Beispiele:

T#1D2h = 1 Tag und 2 Stunden
t#26H = 26 Stunden
t#5m45s = 5 Minuten und 45 Sekunden
t#26S200MS = 26 Sekunden und 200 Millisekunden
T#900ms = 900 Millisekunden

Uhrzeit

Mit dem Uhrzeit-Datentyp (TOD) wird eine bestimmte Zeit angegeben, IEC-1131-3 verwendet das Format TOD#HH:MM:SS.ms um anzugeben, daß die nachfolgenden Zeichen eine Uhrzeit darstellen. Beispiel: TOD#23:59:59.999

TMR

Mit dem Datentyp Timer (TMR) wird ein Zeitglied in strukturiertem Text implementiert

TOD

Element der Datentypgruppe ANY_DATE. TOD (Uhrzeit) Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_DATE oder TOD akzeptieren.

Transition

Graphisches Element in einer Ablaufkette, das einen Booleschen Wert ergibt . Dieses Boolesche Ergebnis legt fest, wann der Programmfluß von den Schritten vor der Transition zu den Schritten nach der Transition weitergeleitet wird.

Transitionsbedingung

Eine Transitionsbedingung ist ein logischer Ausdruck, der mit einem AS-Transitionselement verknüpft ist und einen einzelnen Booleschen Wert ergibt. Dieses Boolesche Ergebnis legt fest, wann die Aktivierung vom aktiven Schritt zu den Schritten nach der Transition weitergeleitet wird.

Transitionslogik

Die Transitionslogik kann von einem einzelnen Booleschen Ausdruck oder einem einzelnen Kontaktplan-Strompfad dargestellt werden.

Transitionsmodus

Mit der Schaltfläche Transitionsmodus auf der AS-Symbolleiste wird festgelegt, welcher Transitionstyp mit dem Transitions-Einfügewerkzeug eingefügt wird. Ist die Schaltfläche Transitionsmodus nicht gedrückt, wird eine Kontaktplantransition eingefügt. Ist die Schaltfläche Transitionsmodus gedrückt, wird eine Boolesche Transition eingefügt.

UINT

Element der Datentypgruppe ANY_INT. UINT Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_INT oder UINT akzeptieren. Der Datentyp UINT stellt eine ganze Zahl ohne Vorzeichen dar, die aus einer oder mehreren Ziffern (0-9) besteht und keinen Dezimalpunkt enthalten darf. Ein UINT ist 16 Bits lang und hat einen Bereich von 0 bis 65535.

WORD

Element der Datentypgruppe ANY_BIT. WORD Datentypen sind zulässig in allen Anweisungen und Funktionsblöcken, die die Datentypen ANY, ANY_BIT oder WORD akzeptieren. Der Datentyp WORD stellt eine ganze Zahl ohne Vorzeichen dar, die aus einer oder mehreren Ziffern (0-9) besteht und keinen Dezimalpunkt enthalten darf. Ein WORD ist 16 Bits lang und hat einen Bereich von 0 bis 65535.